```
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III      VVV         VVV  EEE              RRR          RRR
DDD          DDD RRRRRRRRRRRR           III      VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD          DDD RRRRRRRRRRRR           III      VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD          DDD RRRRRRRRRRRR           III      VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD          DDD RRR   RRR              III      VVV         VVV  EEE              RRR   RRR
DDD          DDD RRR   RRR              III      VVV         VVV  EEE              RRR   RRR
DDD          DDD RRR   RRR              III      VVV         VVV  EEE              RRR   RRR
DDD          DDD RRR       RRR          III        VVV     VVV    EEE              RRR       RRR
DDD          DDD RRR       RRR          III        VVV     VVV    EEE              RRR       RRR
DDD          DDD RRR       RRR          III        VVV     VVV    EEE              RRR       RRR
DDDDDDDDDDDD     RRR           RRR  IIIIIIIII         VVV       EEEEEEEEEEEEEEE  RRR           RRR
DDDDDDDDDDDD     RRR           RRR  IIIIIIIII         VVV       EEEEEEEEEEEEEEE  RRR           RRR
DDDDDDDDDDDD     RRR           RRR  IIIIIIIII         VVV       EEEEEEEEEEEEEEE  RRR           RRR
```

FILEID**TSDRIVER

```
TTTTTTTTTT    SSSSSSSS  DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
TTTTTTTTTT    SSSSSSSS  DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
   TT        SS        DD      DD  RR      RR      II      VV      VV  EE          RR      RR
   TT        SS        DD      DD  RR      RR      II      VV      VV  EE          RR      RR
   TT        SS        DD      DD  RR      RR      II      VV      VV  EE          RR      RR
   TT        SS        DD      DD  RR      RR      II      VV      VV  EE          RR      RR
   TT          SSSSSS  DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
   TT          SSSSSS  DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
   TT                SS DD      DD  RR  RR          II      VV      VV  EE          RR  RR
   TT                SS DD      DD  RR  RR          II       VV    VV   EE          RR  RR
   TT                SS DD      DD  RR    RR        II        VV  VV    EE          RR    RR      ....
   TT                SS DD      DD  RR    RR        II        VV  VV    EE          RR    RR      ....
   TT        SSSSSSSS   DDDDDDDD    RR      RR    IIIIII       VV     EEEEEEEEEE  RR      RR    ....
   TT        SSSSSSSS   DDDDDDDD    RR      RR    IIIIII       VV     EEEEEEEEEE  RR      RR    ....

LL              IIIIII     SSSSSSSS
LL              IIIIII     SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE TSDRIVER - VAX/VMS TS11/TS04 MAGTAPE SUBSYSTEM DRIVER
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28
0000    29  ; E. E. OUYANG 2-APR-79
0000    30  ;
0000    31  ; MODIFIED BY:
0000    32  ;
0000    33  ;       V03-017 MMD0317         Meg Dumont,     25-Jul-1984  11:13
0000    34  ;               Add support for the UCB$L_MEDIA_ID field
0000    35  ;
0000    36  ;       V03-016 MMD0304         Meg Dumont,     27-Jun-1984  15:24
0000    37  ;               Fix to 296 so that only READ REVERSE into BOT returns ENDOFFILE
0000    38  ;
0000    39  ;       V03-015 MMD0296         Meg Dumont,     3-May-1984  9:45
0000    40  ;               Fix to fix MMD0265 we really must return SS$_NORMAL not
0000    41  ;               anyother error code.
0000    42  ;
0000    43  ;       V03-014 ROW0355         Ralph O. Weber          30-APR-1984
0000    44  ;               Modify processing of the IO$M_OPPOSITE modifier so that its
0000    45  ;               use is limited to IO$_REREADN and IO$_REREADP functions by
0000    46  ;               code, rather than by comments.  This provides some protection
0000    47  ;               against accidential misuse of the IO$M_CLSEREXCP bit which is
0000    48  ;               relivant only for tape class driver tapes but which shares the
0000    49  ;               same modifier bit as IO$M_OPPOSITE.
0000    50  ;
0000    51  ;       V03-013 RAS0300         Ron Schaefer            27-Apr-1984
0000    52  ;               Add DEV$M_NNM characteristic to DECHAR2 so that these
0000    53  ;               devices will have the "node$" prefix.
0000    54  ;
0000    55  ;       V03-012 MMD0265         Meg Dumont,     22-Mar-1984  15:28
0000    56  ;               Fix so that reverse into BOT returns SS$_ENDOFFILE like other
0000    57  ;               drivers.
```

```
0000     58 ;
0000     59 ;       V03-011 MMD0225         Meg Dumont,     23-Jan-1984  11:27
0000     60 ;               Deleted the check in the drivers' unit init routine which
0000     61 ;               checked on powerfail to see if the TS SUBSYSTEM was ready
0000     62 ;               before reloading registers etc.. This check was no
0000     63 ;               longer necessary since Robert added the code TEST_NBA which
0000     64 ;               makes sure the controller is available before we allow
0000     65 ;               the QIO to start on the device.
0000     66 ;
0000     67 ;       V03-010 MMD0219         Meg Dumont,     9-Jan-1984  13:59
0000     68 ;               Instead of checking for powerfail at TS_INIT check
0000     69 ;               for command buffer allocated. Fix for support of
0000     70 ;               switchable unibus
0000     71 ;
0000     72 ;       V03-009 ROW0258         Ralph O. Weber          17-NOV-1983
0000     73 ;                       The Paul Painter Memorial Enhancement
0000     74 ;               Named for one of the unfortunate customers who suffered much
0000     75 ;               to determine the great UCB$L_MT_RECORD secret while trying to
0000     76 ;               create a user-written magtape driver, this change eliminates
0000     77 ;               use of the device dependent field, UCB$L_MS_RECORD in favor of
0000     78 ;               the device independent field, UCB$L_RECORD.
0000     79 ;
0000     80 ;       V03-008 ROW0213         Ralph O. Weber          20-AUG-1983
0000     81 ;               Change basing for device-dependent UCB from UCB$L_DP_LINK+4 to
0000     82 ;               a field independent UCB$K_LCL_TAPE_LENGTH.  This allows the
0000     83 ;               device-indpendent UCB to be altered without having to edit
0000     84 ;               this module.
0000     85 ;
0000     86 ;       V03-007 BLS0234         Benn Schreiber          9-Aug-1983
0000     87 ;               Use general addressing mode for EXE$READ_TODR.
0000     88 ;
0000     89 ;       V03-006 KDM0060         Kathleen D. Morse       14-Jul-1983
0000     90 ;               Change references to IPR TODR to use cpu-dependent
0000     91 ;               routine, EXE$READ_TODR.
0000     92 ;
0000     93 ;       V03-005 RLRDPATH1       Robert L. Rappaport     31-May-1983
0000     94 ;               Allow UCB to include new DUAL PORT extension by
0000     95 ;               changing base of where we begin the private TSDRIVER
0000     96 ;               extension from UCB$L_DPC+4 to UCB$L_DP_LINK+4.
0000     97 ;
0000     98 ;       V03-004 RLRTRACE        Robert L. Rappaport     11-Feb-1983
0000     99 ;               Add conditionally assembled trace facility.
0000    100 ;
0000    101 ;       V03-003 RLR52135        Robert L. Rappaport     22-Dec-1982
0000    102 ;               Prevent reverse into BOT from returning SS$_OPINCOMPL.
0000    103 ;
0000    104 ;       V03-002 RLR0001         Robert L. Rappaport     15-July-1982
0000    105 ;               Prevent logging two errors for each soft retry.
0000    106 ;
0000    107 ;       V03-001 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    108 ;               Added $DCDEF, $DEVDEF, $DYNDEF, $PRDEF and $VADEF.
0000    109 ;
0000    110 ;
0000    111 ; TS11/TS04 MAGTAPE DRIVER
0000    112 ;
0000    113 ; MACRO LIBRARY CALLS
0000    114 ;
```

```
0000    115
0000    116          $CRBDEF                         ;DEFINE CRB OFFSETS
0000    117          $DCDEF                          ;DEFINE DEVICE TYPES
0000    118          $DDBDEF                         ;DEFINE DDB OFFSETS
0000    119          $DEVDEF                         ;DEFINE DEVICE TYPES
0000    120          $DPTDEF                         ;DEFINE DPT OFFSETS
0000    121          $DYNDEF                         ;DEFINE DYNAMIC DATA STRUCTURE TYPES
0000    122          $EMBDEF                         ;DEFINE EMB OFFSETS
0000    123          $IDBDEF                         ;DEFINE IDB OFFSETS
0000    124          $IODEF                          ;DEFINE I/O FUNCTION CODES
0000    125          $IRPDEF                         ;DEFINE IRP OFFSETS
0000    126          $MTDEF                          ;DEFINE MAGTAPE STATUS BITS
0000    127          $PRDEF                          ;DEFINE PROCESSOR REGISTERS
0000    128          $SSDEF                          ;DEFINE QIO STATUS RETURN CODES
0000    129          $UCBDEF                         ;DEFINE UCB OFFSETS
0000    130          $VADEF                          ;DEFINE VIRTUAL ADDRESS FIELDS
0000    131          $VECDEF                         ;DEFINE INTERRUPT DISPATCH VECTOR OFFSETS
0000    132          $WCBDEF                         ;DEFINE WCB OFFSETS
0000    133
0000    134 ;
0000    135 ; LOCAL MACROS
0000    136 ;
0000    137 ; EXECUTE HARDWARE COMMAND AND BRANCH ON RETRIABLE ERROR CONDITION
0000    138 ;
0000    139
0000    140          .MACRO  EXHC    BDST,HC
0000    141                  .IF NB  HC
0000    142                  MOVZBL  #CD'HC,R0        ;GET HARDWARE COMMAND INDEX
0000    143                  .ENDC
0000    144                  BSBW    HCEX             ;CALL HARDWARE COMMAND EXECUTION ROUTINE
0000    145                  .WORD   BDST-.-2         ;BRANCH ADDR. ON ERROR CONDITION
0000    146          .ENDM   EXHC
0000    147
0000    148 ; MACRO TO CALL G^IOC$LOADUBAMAPA
0000    149
0000    150          .MACRO  LOADUBAA
0000    151          JSB     G^IOC$LOADUBAMAPA
0000    152          .ENDM   LOADUBAA
0000    153
0000    154
0000    155 ;
0000    156 ; GENERATE HARDWARE COMMAND TABLE ENTRY AND CASE TABLE INDEX SYMBOL
0000    157 ;
0000    158
0000    159          .MACRO  GENHC   HC
0000    160                  CD'HC=<.-HCTAB>/2        ;DEFINE TABLE INDEX SYMBOL
0000    161                  .WORD   HC               ;HARDWARE COMMAND TABLE ENTRY
0000    162          .ENDM   GENHC
0000    163
0000    164
0000    165 ;
0000    166 ; LOCAL SYMBOLS
0000    167 ;
0000    168 ;
0000    169 ; TS11/TS04 COMMAND PACKET DEFINITION
0000    170 ;
0000    171 ;
```

```
                                        E 3
                     0000   172
                     0000   173          $DEFINI MS
                     0000   174
                     0000   175
00000000             0000   176 .=0                                ;RESET PC??????
                     0000   177 $DEF     MS_CPHD .BLKW   1          ;COMMAND PACKET HEADER
                     0002   178          _VIELD  MS_CPHD,0,<-       ;
                     0002   179                  <COD,5>,-          ;COMMAND CODE FIELD
                     0002   180                  <,2>,-             ;B5-B6 ALWAYS 0 FOR TS04
                     0002   181                  <IE,,M>,-          ;INTERRUPT ENABLE
                     0002   182                  <MOD,4>,-          ;COMMAND MODE FIELD(B11-B8)
                     0002   183                                     ; B8=REVERSE & B9=RETRY
                     0002   184                  <SWB,,M>,-         ;SWAP BYTES BIT(B12)
                     0002   185                  <OPP,,M>,-         ;OPPOSITE BIT(B13)
                     0002   186                  <CVC,,M>,-         ;CLEAR VOLUME CHECK(B14)
                     0002   187                  <ACK,,M>,-         ;ACKNOWLEDGE BIT(B15)
                     0002   188          >                          ;
                     0002   189 $DEF     MS_BACT .BLKW   1          ;BUS ADDRESS(B15-B0) OR COUNT
                     0004   190 $DEF     MS_BA1  .BLKW   1          ;BUS ADDRESS B17-B16(RIGHT JUST)
                     0006   191 $DEF     MS_CNT  .BLKW   1          ;BYTE COUNT
                     0008   192                                     ;FOR WRITE CHARACTERISTIC DATA
                     0008   193 $DEF     MS_MBA0 .BLKW   1          ;MESSAGE BUFFER ADDR. WRD 1
                     000A   194 $DEF     MS_MBA1 .BLKW   1          ;MESSAGE BUFFER ADDR. WRD 2
                     000C   195 $DEF     MS_LNTH .BLKW   1          ;MESSAGE BUFFER LENGTH(ALWAYS 14.)
                     000E   196 $DEF     MS_CHWD .BLKW   1          ;CHARACTERISTIC WORD
                     0010   197          _VIELD  MS_CHWD,4,<-       ;
                     0010   198                  <ERI,,M>,-         ;ENABLE MESSAGE BUFFER RELEASE INTERRUPTS
                     0010   199                  <EAI,,M>,-         ;ENABLE ATTENTION INTERRUPTS
                     0010   200                  <ENB,,M>,-         ;USED WITH ESS BIT***
                     0010   201                  <ESS,,M>,-         ;ENABLE SKIP TAPE MARKS STOP
                     0010   202          >                          ;
                     0010   203
                     0010   204 ;
                     0010   205 ; TS11/TS04 MESSAGE PACKET DEFINITION
                     0010   206 ;
                     0010   207
                     0010   208
                     0010   209 $DEF     MS_MHD  .BLKW   1          ;MESSAGE PACKET
                     0012   210          _VIELD  MS_MHD,0,<-        ;MESSAGE HEADER WORD
                     0012   211                  <COD,5>,-          ;MESSAGE CODE FIELD
                     0012   212                  <FMT,3>,-          ;FORMAT FIELD
                     0012   213                  <CLS,4>,-          ;CLASS CODE FIELD
                     0012   214                  <RSR,3>,-          ;RESERVED FIELD
                     0012   215                  <ACK,,M>,-         ;MESSAGE ACKNOWLEDGE BIT(B15)
                     0012   216          >                          ;
                     0012   217 $DEF     MS_LNH  .BLKW   1          ;MESSAGE LENGTH WORD
                     0014   218                                     ;HIGH BYTE=0,LOW BYTE=1010(LENGTH)
                     0014   219 $DEF     MS_RBPC .BLKW   1          ;RESIDUAL BYTE/POSITION COUNT
                     0016   220 $DEF     MS_XSR0 .BLKW   1          ;EXTENDED STATUS REGISTER 0
                     0018   221          _VIELD  MS_XSR0,0,<-       ;
                     0018   222                  <EOT,,M>,-         ;END OF TAPE DETECTED(B0)
                     0018   223                  <BOT,,M>,-         ;BEGINNING OF TAPE(B1)
                     0018   224                  <WLK,,M>,-         ;WRITE LOCKED(B2)
                     0018   225                  <PED,,M>,-         ;PHASE ENCODED DRIVE(B3)
                     0018   226                  <VCK,,M>,-         ;VOLUME CHECK(B4)
                     0018   227                  <IE,,M>,-          ;INTERRUPT WAS ENABLED(B5)
                     0018   228                  <ONL,,M>,-         ;DEVICE ON-LINE(B6)
```

```
0018   229                  <MOT,,M>,-              ;TAPE MOVING ON LAST COMMAND(B7)
0018   230                  <ILA,,M>,-              ;ILLEGAL ADDRESS(B8)
0018   231                  <ILC,,M>,-              ;ILLEGAL COMMAND(B9)
0018   232                  <NEF,,M>,-              ;NON-EXECUTALBE FUNCTION(B10)
0018   233                  <WLE,,M>,-              ;WRITE LOCK ERROR(B11)
0018   234                  <RLL,,M>,-              ;RECORD LENGTH LONG(B12)
0018   235                  <LET,,M>,-              ;LOGICAL END OF TAPE(B13)
0018   236                  <RLS,,M>,-              ;RECORD LENGTH SHORT(B14)
0018   237                  <TMK,,M>,-              ;TAPE MARK DETECTED(B15)
0018   238              >                           ;
0018   239 $DEF     MS_XSR1 .BLKW   1               ;EXTENDED STATUS REGISTER 1
001A   240             _VIELD  MS_XSR1,0,<-         ;
001A   241                  <MTE,,M>,-              ;(PE) MULTI-TRACK ERROR
001A   242                          -               ;(NRZ) VERTICAL PARITY ERROR
001A   243                  <UNC,,M>,-              ;(PE) UNCORRECTABLE DATA ERROR(B1)
001A   244                          -               ;(NRZ) CYCLIC REDUNDANCY CHECK ERROR
001A   245                  <POL,,M>,-              ;(PE) POSTAMBLE LONG(B2)
001A   246                          -               ;(NRZ) LONGITUDINAL REDUNDANCY CHECK ERROR
001A   247                  <POS,,M>,-              ;(PE) POSTAMBLE SHORT(B3)
001A   248                          -               ;(NRZ) NOISE RECORD
001A   249                  <IED,,M>,-              ;(PE) INVALID END DATA(B4)
001A   250                          -               ;(NRZ) LRC WAS 0.
001A   251                  <IPO,,M>,-              ;(PE) INVALID POSTAMBLE(B5)
001A   252                          -               ;(NRZ) ILLEGAL TAPE MARK
001A   253                  <SYN,,M>,-              ;(PE) SYNCH ERROR(B6)
001A   254                          -               ;(NRZ) FRAME DROPOUT
001A   255                  <IPR,,M>,-              ;(PE) INVALID PREAMBLE(B7)
001A   256                  <,1>,-                  ;RESERVED BIT
001A   257                  <SCK,,M>,-              ;SPEED CHECK(B9)
001A   258                  <DBF,,M>,-              ;(PE) DESKEW BUFFER FAIL(B10)
001A   259                          -               ;(NRZ) NRZ BOARD FIFO OVERFLOW
001A   260                  <TIG,,M>,-              ;TRASH IN GAP(B11)
001A   261                  <CRS,,M>,-              ;CREASE DETECTED(B12)
001A   262                  <COR,,M>,-              ;CORRCTABLE DATA(B13)
001A   263                  <,1>,-                  ;UNUSED BIT(B14)
001A   264                  <DLT,,M>,-              ;DATA LATE(B15)
001A   265              >
001A   266 $DEF     MS_XSR2 .BLKW   1               ;EXTENDED STATUS REGISTER 2
001C   267             _VIELD  MS_XSR2,0,<-         ;
001C   268                  <DTP,8>,-               ;DEAD TRACK PARITY,B7-B0
001C   269                  <XSK,,M>,-              ;EXCESSIVE SKEW(B9)
001C   270                  <WCF,,M>,-              ;WRITE CLOCK FAIL(B10),BROKEN HARDWARE
001C   271                  <,1>,-                  ;B11 NOT USED
001C   272                  <CAF,,M>,-              ;CAPSTAN ACCELERATION FAIL(B12)
001C   273                  <BPE,,M>,-              ;SERIAL BUS PARITY ERROR AT DRIVE(B13)
001C   274                  <SIP,,M>,-              ;SILO PARITY ERROR(B14)
001C   275                  <OPM,,M>,-              ;OPERATION IN PROGRESS(B15)
001C   276              >
001C   277 $DEF     MS_XSR3 .BLKW   1               ;EXTENDED STATUS REGISER 3
001E   278             _VIELD  MS_XSR3,0,<-         ;
001E   279                  <RIB,,M>,-              ;REVERSE INTO BOT(B0)
001E   280                  <LXS,,M>,-              ;LIMIT EXCEEDED STATICALLY(B1)
001E   281                  <NOI,,M>,-              ;NOISE RECORD(B2)
001E   282                  <DCK,,M>,-              ;DENSITY CHECK(B3)
001E   283                  <CRF,,M>,-              ;CAPSTAN RESPONSE FAIL(B4)
001E   284                  <REV,,M>,-              ;TAPE MOVED BACKWARDS(B5)
001E   285                  <OPI,,M>,-              ;OPERATION IN COMPLETE(B6)
```

```
                001E   286                 <LMX,,M>,-              ;TAPE LIMIT EXCEEDED(B7)
                001E   287                 <FEC,8>,-               ;B15-B8, FATAL ERROR CODE(U-DIAGNOSTIC)
                001E   288         >
                001E   289
                001E   290         $DEFEND MS
                0000   291
6CE9300B        0000   292 MEDIA_ID_TS11 = ^X<6CE9300B>
                0000   293 ;
                0000   294 ; TS11/TS04 TSSR TERMINATION CLASS CODES
                0000   295 ;
                0000   296
00000000        0000   297 TCC_NML=0                               ;NORAML TERMINATION
00000001        0000   298 TCC_ATN=1                               ;ATTENTION CONDITION
00000002        0000   299 TCC_TSA=2                               ;TAPE STATUS ALERT
00000003        0000   300 TCC_FNR=3                               ;FUNCTION REJECT
00000004        0000   301 TCC_REM=4                               ;RECOVERABLE ERROR(TAPE MOVED)
00000005        0000   302 TCC_REN=5                               ;RECOVERABLE ERROR(TAPE NOT MOVED)
00000006        0000   303 TCC_UER=6                               ;UNRECOVERABLE ERROR(TAPE POSI. LOST)
00000007        0000   304 TCC_FTL=7                               ;FATAL CONTROLLER ERROR
                0000   305
                0000   306 ;
                0000   307 ; FATAL CLASS (FC) CODES IN TSSR
                0000   308 ;
                0000   309
00000000        0000   310 FCC_IDF=0                               ;INTERNAL DIAG. FAILURE
00000001        0000   311 FCC_CPE=1                               ;IO SEQUENCE CROM PARITY ERROR
00000002        0000   312 FCC_UPE=2                               ;U-PROCESSOR CROM PARITY ERROR OR OTHER
00000003        0000   313 FCC_LAP=3                               ;LOSS OF AC POWER DETECTED
                0000   314
                0000   315 ;
                0000   316 ; TS11/TS04 MESSAGE CODES IN MS_MHD_COD
                0000   317 ;
                0000   318
00000010        0000   319 MSG_END=^O020                           ;END
00000011        0000   320 MSG_FAL=^O021                           ;FAIL
00000012        0000   321 MSG_ERR=^O022                           ;ERROR
00000013        0000   322 MSG_ATN=^O023                           ;ATTENTION
00000014        0000   323 MSG_LOG=^O024                           ;LOG (NOT USED)
                0000   324
                0000   325 ;
                0000   326 ; CLASS CODE FOR MESSAGE CODES (MS_MHD_CLS VALUES)
                0000   327 ;
                0000   328
                0000   329                                 ;**WHEN MSG TYPE=ATTENTION**
00000000        0000   330 CLS_ONF=0                               ;ON OR OFFLINE
00000001        0000   331 CLS_MDF=1                               ;MICRO DIAG. FAILURE
                0000   332                                 ;**WHEN MSG TYPE=FAIL**
00000000        0000   333 CLS_PTB=0                               ;PACKET BAD(SERIAL BUS PARITY ERROR)
00000001        0000   334 CLS_OTHER=1                             ;OTHERS
00000002        0000   335 CLS_WLN=2                               ;WRITE LOCK ERROR OR NON-EXECUTABLE FUNCTION
00000003        0000   336 CLS_MDE=3                               ;MICRO DIAGNOSTIC ERROR
                0000   337
                0000   338 ;
                0000   339 ; TS11/TS04 HARDWARE COMMAND MODES/CODES
                0000   340 ;
                0000   341
                0000   342                                       ; INTERRUPT ENABLE & ACKNOWLEDGE
```

```
00000000  0000   343 HC_NOP=0                          ;SIMULATED NOP(REAL NO OPERATION)
00000000  0000   344 HC_PAK=HC_NOP                     ;SIMULATED PACK ACKNOWLEDGE
00000000  0000   345 HC_WCK=HC_NOP                     ;SIMULATED WRITE CHECK
00000000  0000   346 HC_WKR=HC_NOP                     ;SIMULATED WRITE CHECK REVERSE
00000000  0000   347 HC_RPS=HC_NOP                     ;SIMULATED READ IN PRESET
00000000  0000   348 HC_SCH=HC_NOP                     ;SIMULATED SET CHARACTERISTICS
          0000   349
          0000   350
0000C081  0000   351 HC_RDN=^O00001!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;* READ NEXT (FORWARD)
0000C181  0000   352 HC_RDP=^O00401!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;* READ PREVIOUS (REVERSE)
0000C281  0000   353 HC_RRP=^O01001!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;* REREAD PREVIOUS (SPACE RE
0000C381  0000   354 HC_RRN=^O01401!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;* REREAD NEXT (SPACE FWD, R
0000C084  0000   355 HC_WRC=^O00004!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;+ WRITE CHARACTERISTICS
000CC085  0000   356 HC_WRD=^O00005!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;* WRITE DAT
0000C285  0000   357 HC_WDR=^O01005!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;* WRITE DATA RETRY (SPACE R
          0000   358                                                          ; WRITE DATA)
0000C086  0000   359 HC_WSM=^O00006!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;* WRITE SUBSYSTEM MEMORY
0000C088  0000   360 HC_SRF=^O00010!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;$ SPACE RECORDS FORWARD
0000C188  0000   361 HC_SRR=^O00410!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;$ SPACE RECORDS REVERSE
          0000   362 ;HC_STF=^O01010!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK ;$ SKIP TAPE MARKS FORWARD
          0000   363 ;HC_STR=^O01410!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK ;$ SKIP TAPE MARKS REVERSE
          0000   364  ;**NOTE** SKIP TAPE MARK COMMANDS ARE SIMULATED BY SKIP RECORD COMMANDS***
0000C088  0000   365 HC_STF=^O00010!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;$ SPACE TAPE MARK FORWARD
0000C188  0000   366 HC_STR=^O00410!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;$ SPACE TAPE MARKS REVERSE
0000C488  0000   367 HC_RWD=^O02010!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;$ REWIND
0000C089  0000   368 HC_WTM=^O00011!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;- WRITE TAPE MARK
0000C189  0000   369 HC_ERS=^O00411!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;- ERASE
0000C289  0000   370 HC_WTR=^O01011!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;- WRITE TAPE MARK RETRY (SP
          0000   371                                                          ; ERASE, WRTIE TAPE MARK)
0000C08A  0000   372 HC_BRL=^O00012!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;- MESSAGE BUFFER RELEASE
0000C18A  0000   373 HC_UNL=^O00412!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;- REWIND AND UNLOAD
0000C28A  0000   374 HC_CLN=^O01012!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;- CLEAN
0000C08B  0000   375 HC_DRI=^O00013!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;- DRIVER INITIALIZE
0000C08F  0000   376 HC_GST=^O00017!MS_CPHD_M_IE!MS_CPHD_M_CVC!MS_CPHD_M_ACK  ;- GET STATUS IMMEDIATE
          0000   377                                ;**NOTE**
          0000   378                                ; * => DATA XFR
          0000   379                                ; + => (SPECIAL)
          0000   380                                ; $ => POSITION
          0000   381                                ; - => FORMAT,CONTROL,INITIALIZE,& STATUS
          0000   382
          0000   383 ;
          0000   384 ; DEFINE DEVICE DEPENDENT UNIT CONTROL BLOCK OFFSETS
          0000   385 ;
          0000   386
          0000   387         $DEFINI UCB
          0000   388
          0000   389         $VIELD  UCB,0,<-                  ;DEV. DEP. STATUS BITS IN UCB$W_DEVSTS
          0000   390                 <MS_FEF,,M>,-             ;TAPE IS PAST ONE TAPE MARK
          0000   391                 <MS_SWAP,,M>,-            ;*SWAP BYTES FOR COMPATIBILITY MODE
          0000   392                 <MS_IWR,,M>,-             ;*INHIBIT WRITE RETRIES
          0000   393                 <MS_SER,,M>,-             ;SELECT ERROR HAS OCCURRED????
          0000   394                 <MS_RWD,,M>,-             ;UNIT IS REWINDING
          0000   395                 <MS_RDPR,,M>,-            ;REQUEST DATAPATH FLAG
          0000   396                 <MS_SWE,,M>,-             ;DOING SOFTWARE EMULATION
          0000   397                 <MS_NER,,M>,-             ;NO ERROR RECOVERY
          0000   398                 <MS_UMD,,M>,-             ;USER MODE DIAGNOSTIC REQUEST
          0000   399                 <MS_RSP,,M>,-             ;REWIND/SPACE IN PROGRESS
```

```
                     0000   400                <MS_LBA,,M>,-           ;LOADING BUFFER ADDR. INTO TS04
                     0000   401                <MS_RPI,,M>,-           ;RPOSITIONING IN PROGRESS
                     0000   402                <MS_VCK,,M>,-           ;VOLUME CHECK
                     0000   403                <MS_RIP,,M>,-           ;RETRY IN PROGRESS FLAG
                     0000   404            >
                     0000   405
                     0000   406            ;** STATUS BITS DEFINED ELSWHERE***
                     0000   407            ;** IN UCB$L_DEVDEPEND:
                     0000   408            ;**     MT$M_PARITY=1, IF EVEN;0, IF ODD
                     0000   409            ;**     MT$V_FORMAT=MT$K_DEFAULT/NORMAL11/CORDMP11/NORMAL15
                     0000   410            ;**     MT$V_DENSITY=MT$K_PE_1600/MT$K_NRZI-800
                     0000   411            ;**     MT$M_BOT=TAPE IS AT BOT
                     0000   412            ;**     MT$M_EOF=TAPE AT EOF
                     0000   413            ;**     MT$M_EOT=TAPE AT EOT
                     0000   414            ;**     MT$M_HWL=HARDWARE WRITE LOCKED
                     0000   415            ;**     MT$M_LOST=TAPE POSITION LOST
                     0000   416            ;** IN UCB$W_STS:
                     0000   417            ;**     UCB$M_TIM=TIMEOUT ENABLED
                     0000   418            ;**     UCB$M_INT=INTERRUPT EXPECTED
                     0000   419            ;**     UCB$M_ERLOGIP=ERRORLOG IN PROGRESS
                     0000   420            ;**     UCB$M_CANCEL=CANCEL I/O
                     0000   421            ;**     UCB$M_ONLINE=UNIT ONLINE
                     0000   422            ;**     UCB$M_POWER=POWER FAILED WHILE UNIT BUSY
                     0000   423            ;**     UCB$M_TIMOUT=UNIT TIME OUT
                     0000   424            ;**     UCB$M_INTTYPE=RECEIVER INTERRUPT,IF SET
                     0000   425            ;**     UCB$M_BSY=UNIT IS BUSY
                     0000   426            ;**     UCB$M_MOUNTING=DEVICE IS BEING MOUNTED
                     0000   427            ;**     UCB$M_DEADMO=DEALLOCATE AT DISMOUNT
                     0000   428            ;**     UCB$M_VALID=VOLUME IS SOFTWARE VALID
                     0000   429            ;**     UCB$M_UNLOAD=UNLOAD VOLUME AT DISMOUNT
                     0000   430            ;** IN UCB$L_DEVCHAR:
                     0000   431            ;** .......
                     0000   432            ;**     DEV$M_SWL=SOFTWARE WRITE LOCKED
                     0000   433            ;** .......
                     0000   434            ;
                     0000   435            ; NEW EXTENSION TO UCB FOR TS11/TS04
                     0000   436            ;
                     0000   437
00000084             0000   438            .=UCB$K_LCL_TAPE_LENGTH
                     00B4   439 $DEF        UCB$W_MS_SPACNT .BLKW   1       ;SPACING COUNT
                     00B6   440 $DEF        UCB$L_MS_TSPT1  .BLKL   1       ;PTR. TO TS04 BUFFER IN
                     00BA   441                                             ;NON-PAGED POOL
                     00BA   442 $DEF        UCB$L_MS_TSPT2  .BLKL   1       ;CORRESPONDING UNIBUS ADDR.
                     00BE   443 $DEF        UCB$W_MS_TSPT3  .BLKW   1       ;COMMAND PTR FOR TS11/TS04
                     00C0   444 $DEF        UCB$W_MS_TSBA   .BLKW   1       ;TS11/TS04 DEVICE REGISTER(TSBA)
                     00C2   445 $DEF        UCB$W_MS_TSSR   .BLKW   1       ;TS11/TS04 DEVICE REGISTER(TSSR)
                     00C4   446                _VIELD  MS_TSSR,1,<-          ;TS11/TS04 STATUS REGISTER(B0 UNUSED)
                     00C4   447                        <TCC,3>,-             ;TERMINATION CLASS CODE FIELD
                     00C4   448                        <FC,2>,-              ;FATAL ERROR CLASS CODE FIELD
                     00C4   449                        <OFL,,M>,-            ;DEVICE IS OFF-LINE(B6)
                     00C4   450                        <SSR,,M>,-            ;SUBSYSTEM READY(B7)
                     00C4   451                        <A16,,M>,-            ;BUFFER ADDRESS BIT 16
                     00C4   452                        <A17,,M>,-            ;BUFFER ADDRESS BIT 17
                     00C4   453                        <NBA,,M>,-            ;NEED BUFFER ADDRESS(B10)
                     00C4   454                        <NXM,,M>,-            ;NON-EXISTENT MEMORY(B11)
                     00C4   455                        <RMR,,M>,-            ;REGISTER MODIFICATION REFUSED(B12)
                     00C4   456                        <SPE,,M>,-            ;SERIAL BUS PARITY ERROR(B13)
```

```
                     00C4   457                     <UPE,,M>,-              ;UNIBUS PARITY ERROR(B14)
                     00C4   458                     <SC,,M>,-               ;SPECIAL CONDITION(B15)
                     00C4   459             >                               ;
                     00C4   460                                     ;**FATAL ERROR CONDITION: UPE!SPE!NXM!NBA
                     00C4   461 $DEF     UCB$W_MS_XC         .BLKW   1       ;BYTES XFERRED OR RECORDS/FILES SKIPPED
                     00C6   462 $DEF     UCB$B_MS_DPN        .BLKB   1       ;DATA PATH NUMBER
                     00C7   463 $DEF     UCB$B_MS_PER        .BLKB   1       ;PURGE ERROR IF BIT 0 SET
                     00C8   464 $DEF     UCB$L_MS_DPR        .BLKL   1       ;DATA PATH REGISTER USED
                     00CC   465 $DEF     UCB$L_MS_FMPR       .BLKL   1       ;FINAL MAP REGISTER
                     00D0   466 $DEF     UCB$L_MS_PMPR       .BLKL   1       ;FINAL-1(PREVIOUS) MAP REGISTER
                     00D4   467                                      ;**NOTE**LAST 1 LONGWORD IS USED DURING
                     00D4   468                                      ;****POWERFAIL REPOSITIONING
                     00D4   469 $DEF     UCB$L_MS_NMPR       .BLKL   1       ;FINAL+1(NEXT) MAP REGISTER
                     00D8   470 $DEF     UCB$L_MS_OMPR       .BLKL   1       ;COPY OF VEC$W_MAPREG(LONGWORD IN CRB)
                     00DC   471 $DEF     UCB$L_MS_TIMOUT     .BLKL   1       ; Timout value for function in progress
                     00E0   472 $DEF     UCB$Q_MS_TMP1       .BLKQ   1       ;TEMP FOR UCB$W_BCNT,BOFF, and SVAPTE
                     00E8   473 $DEF     UCB$L_MS_TMP2       .BLKL   1       ;TEMP. FOR CRB$L_INTD+VEC$W_MAPREG
                     00EC   474 $DEF     UCB$Q_MS_BUFSVAPTE                  ;AREA TO SAVE PARAMETERS TO MAP MESSAGE
000000F4             00EC   475                               .BLKQ   1       ; BUFFER IN UNIBUS SPACE
                     00F4   476 $DEF     UCB$L_MS_TPOSITN    .BLKL   1       ;TAPE POSITION AT POWERFAIL
                     00F8   477 $DEF     UCB$W_MS_MHD        .BLKW   1       ;MESSAGE PACKET**COPY IN UCB**
                     00FA   478 $DEF     UCB$W_MS_LNH        .BLKW   1       ;MESSAGE LENGTH WORD
                     00FC   479 $DEF     UCB$W_MS_RBPC       .BLKW   1       ;RESIDUAL BYTE/POSITOIN COUNT
                     00FE   480 $DEF     UCB$W_MS_XSR0       .BLKW   1       ;EXTENDED STATUS REGISTER 0
                     0100   481 $DEF     UCB$W_MS_XSR1       .BLKW   1       ;EXTENDED STATUS REGISTER 1
                     0102   482 $DEF     UCB$W_MS_XSR2       .BLKW   1       ;EXTENDED STATUS REGISTER 2
                     0104   483 $DEF     UCB$W_MS_XSR3       .BLKW   1       ;EXTENDED STATUS REGISTER 3
                     0106   484
                     0106   485             .IF     DF      TS_TRACE
                     0106   486
                     0106   487 $DEF     UCB$W_TRACESTS      .BLKW   ;       ; Status of trace.
                     0106   488 $DEF     UCB$L_TRACEBEG      .BLKL   1       ; Pointer to beginning of trace ring.
                     0106   489 $DEF     UCB$L_TRACEPTR      .BLKL   1       ; Pointer to next available slot.
                     0106   490 $DEF     UCB$L_TRACEND       .BLKL   1       ; Pointer to beyond trace ring.
                     0106   491
                     0106   492 TRACE_V_ACTIVE=0
                     0106   493 TRACE_M_ACTIVE=1
                     0106   494
                     0106   495             .ENDC
00000106             0106   496 UCB$K_MS_LENGTH=.
                     0106   497             $DEFEND UCB
```

```
                                                K 3
0000    499 .SBTTL  DRIVER TABLES
0000    500
0000    501 ;
0000    502 ; DRIVER PROLOGUE TABLE
0000    503 ;
0000    504
0000    505         DPTAB   -                               ;DEFINE DRIVER PROLOGUE TABLE
0000    506                 END=TS_END,-                    ;END OF DRIVER
0000    507                 ADAPTER=UBA,-                   ;UNIBUS ADAPTER
0000    508                 UCBSIZE=UCB$K_MS_LENGTH,-
0000    509                 NAME=TSDRIVER                   ;DRIVER NAME
0038    510         DPT_STORE INIT                          ;CONTROL BLOCK INIT VALUE
0038    511         DPT_STORE DDB,DDB$L_ACPD,L,<^A\MTA\> ;DEFAULT ACP NAME
003F    512         DPT_STORE UCB,UCB$B_FIPL,B,8            ;FORK IPL
0043    513         DPT_STORE UCB,UCB$L_DEVCHAR,L,-         ;DEVICE CHARACTERISTICS
0043    514                 <DEV$M_FOD-                     ; FILES ORIENTED
0043    515                 !DEV$M_DIR-                     ; DIRECTORY STRUCTURED
0043    516                 !DEV$M_AVL-                     ; AVAILABLE
0043    517                 !DEV$M_ELG-                     ; ERROR LOGGING ENABLED
0043    518                 !DEV$M_IDV-                     ; INPUT DEVICE
0043    519                 !DEV$M_ODV-                     ; OUTPUT DEVICE
0043    520                 !DEV$M_SDI-                     ; SINGLE DIRECTORY DEVICE
0043    521                 !DEV$M_SQD>                     ; SEQUENTIAL DEVICE
004A    522         DPT_STORE UCB,UCB$L_DEVCHAR2,L,-        ; DEVICE CHARACTERISTICS
004A    523                 <DEV$M_NNM>                     ; PREFIX NAME WITH "node$"
0051    524         DPT_STORE UCB,UCB$B_DEVCLASS,B,DC$_TAPE ;DEVICE CLASS
0055    525         DPT_STORE UCB,UCB$B_DEVTYPE,B,DT$_TS11  ;DEVICE TYPE
0059    526         DPT_STORE UCB,UCB$L_MEDIA_ID,L,MEDIA_ID_TS11 ;DEVICE MEDIA ID
0060    527         DPT_STORE UCB,UCB$W_DEVBUFSIZ,W,2048 ;DEFAULT BUFFER SIZE
0065    528         DPT_STORE UCB,UCB$L_DEVDEPEND,W,<^X4C0> ;DEFAULT TAPE PARAMETERS
006A    529                                                 ; FORMAT=NORMAL11,DENSITY=1600BPI
006A    530         DPT_STORE UCB,UCB$B_DIPL,B,21           ;DEVICE IPL
006E    531         DPT_STORE UCB,UCB$B_ERTCNT,B,16 ;ERROR RETRY COUNT
0072    532         DPT_STORE UCB,UCB$B_ERTMAX,B,16 ;MAX ERROR RETRY COUNT
0076    533         DPT_STORE REINIT                        ;CONTROL BLOCK RE-INIT VALUES
0076    534         DPT_STORE CRB,CRB$L_INTD+4,D,TS$INT ;INTERRUPT SERVICE ROUTINE ADDR.
007B    535         DPT_STORE CRB,CRB$L_INTD+VEC$L_UNITINIT,D,TS_INIT ;UNIT INIT
0080    536         DPT_STORE DDB,DDB$L_DDT,D,MS$DDT ;DDT ADDRESS
0085    537         DPT_STORE END                           ;
00000001 0000   538         .MDELETE DPT_STORE
0000    539
0000    540 ;
0000    541 ; DRIVER DISPATCH TABLE
0000    542 ;
0000    543
0000    544         DDTAB   MS,-                            ;(MS=GENERIC NAME)DRIVER DISPATCH TABLE
0000    545                 TS_STARTIO,-                    ;START I/O OPERATION
0000    546                 0,-                             ;UNSOLICITED INTERRUPT
0000    547                 TS_FUNCTABLE,-                  ;FUNCTION DECISION TABLE
0000    548                 +IOC$CANCELIO,-                 ;CANCEL I/O ENTRY POINT(STANDARD)
0000    549                 TS_REGDUMP,-                    ;REGISTER DUMP ROUTINE
0000    550                 <8*4+<1+23>*4>,-                        ;DIAG. BUFFER SIZE
0000    551                 <<1+23>*4+EMB$L_DV_REGSAV> ;ERROR BUFFER SIZE
0038    552
0038    553 ;
0038    554 ; HARDWARE COMMAND TABLE - MODES/CODES
0038    555 ;
```

```
0038
0038   HCTAB:
0038            GENHC   HC_NOP                  ;SIMULATED NOP
003A            GENHC   HC_UNL                  ;REWIND & UNLOAD
003C            GENHC   HC_STF                  ;SKIP TAPE MARK FORWARD(SPACE FILE)
003E            GENHC   HC_RWD                  ;REWIND
0040            GENHC   HC_DRI                  ;DRIVE INITIALIZE(DRIVE CLEAR)
0042            GENHC   HC_STR                  ;SKIP TAPE MARK REVERSE
0044            GENHC   HC_ERS                  ;ERASE
0046            GENHC   HC_SRR                  ;SKIP RECORD REVERSE
0048            GENHC   HC_PAK                  ;SIMULATED PACK ACKNOWLEDGE
004A            GENHC   HC_SRF                  ;SKIP RECORD FORWARD
004C            GENHC   HC_WCK                  ;SIMULATED WRITECHECK
004E            GENHC   HC_WRD                  ;WRITE DATA(WRITEPBLK)
0050            GENHC   HC_RDN                  ;READ DATA NEXT(READPBLK)
0052            GENHC   HC_WKR                  ;SIMULATED WRITECHECK REV.
0054            GENHC   HC_WRD                  ;WRITE DATA(NO WRITEPBLK REV.)
0056            GENHC   HC_RDP                  ;READ DATA PREVIOUS
0058            GENHC   HC_RRN                  ;REREAD DATA NEXT
005A            GENHC   HC_RRP                  ;REREAD DATA PREVIOUS
005C            GENHC   HC_WDR                  ;WRITE DATA RETRY
005E            GENHC   HC_RPS                  ;SIMULATED READ PRESET
0060            GENHC   HC_SCH                  ;SIMULATED SET CHARACTERISTIC
0062            GENHC   HC_GST                  ;GET STATUS IMMEDIATE
0064            GENHC   HC_WTM                  ;WRITE TAPE MARK
0066            GENHC   HC_WTR                  ;WRITE TAPE MARK RETRY
0068            GENHC   HC_CLN                  ;CLEAN
006A            GENHC   HC_BRL                  ;MESSAGE BUFFER RELEASE
006C            GENHC   HC_WSM                  ;WRITE SUBSYSTEM MEMORY
006E            GENHC   HC_WRC                  ;WRITE CHARACTERISTIC
0070
0070
```

```
                     0070   589 ;+
                     0070   590
                     0070   591 ; TS11/TS04 FUNCTION DECISION TABLE
                     0070   592 ;-
                     0070   593
                     0070   594 TS_FUNCTABLE:                                ;FUNCTION DECISION TABLE
                     0070   595         FUNCTAB ,-                          ;LEGAL FUNCTIONS
                     0070   596                 <NOP,-                      ;NO OPERATION
                     0070   597                  UNLOAD,-                   ;UNLOAD VOLUME
                     0070   598                  SPACERECORD,-              ;SPACE RECORDS
                     0070   599                  RECAL,-                    ;RECALIBRATE (REWIND)
                     0070   600                  DRVCLR,-                   ;DRIVER CLEAR
                     0070   601                  READPRESET,-               ;READ IN PRESET
                     0070   602                  PACKACK,-                  ;PACK ACKNOWLEDGE
                     0070   603                  ERASETAPE,-                ;ERASE TAPE
                     0070   604                  SENSECHAR,-                ;SENSE TAPE CHARACTERISTICS
                     0070   605                  SETCHAR,-                  ;SET CHARACTERISTICS
                     0070   606                  SPACEFILE,-                ;SPACE FILE
                     0070   607                  WRITECHECK,-               ;WRITE CHECK FORWARD
                     0070   608                  WRITEPBLK,-                ;WRITE PHYSICAL BLOCK
                     0070   609                  WRITERET,-                 ;**NEW**WRITE PHYSICAL BLOCK RETRY
                     0070   610                  READPBLK,-                 ;READ PHYSICAL BLOCK
                     0070   611                  REREADN,-                  ;**NEW**REREAD NEXT
                     0070   612                  REREADP,-                  ;**NEW**REREAD PREVIOUS
                     0070   613                  AVAILABLE,-                ;AVAILABLE (REWIND/NOWAIT CLEAR VALID)
                     0070   614                  WRITEMARK,-                ;WRITE TAPE MARK
                     0070   615                  WRTTMKR,-                  ;**NEW**WRITE TAPE MARK RETRY
                     0070   616                  CLEAN,-                    ;**NEW**CLEAN TAPE
                     0070   617                  READLBLK,-                 ;READ LOGICAL BLOCK
                     0070   618                  WRITELBLK,-                ;WRITE LOGICAL BLOCK
                     0070   619                  SENSEMODE,-                ;SENSE TAPE MODE
                     0070   620                  SETMODE,-                  ;SET MODE
                     0070   621                  REWIND,-                   ;REWIND
                     0070   622                  REWINDOFF,-                ;REWIND AND SET OFFLINE
                     0070   623                  SKIPRECORD,-               ;SKIP RECORDS
                     0070   624                  SKIPFILE,-                 ;SKIP FILES
                     0070   625                  WRITEOF,-                  ;WRITE END OF FILE
                     0070   626                  READVBLK,-                 ;READ VIRTUAL BLOCK
                     0070   627                  WRITEVBLK,-                ;WRITE VIRTUAL BLOCK
                     0070   628                  ACCESS,-                   ;ACCESS FILE AND/OR FIND DIRECTORY
                     0070   629                  ACPCONTROL,-               ;ACP CONNTROL FUNCTION
                     0070   630                  CREATE,-                   ;CREATE FILE AND/OR CREATE DIRECTORY
                     0070   631                  DEACCESS,-                 ;DEACCESS FILE
                     0070   632                  DELETE,-                   ;DELETE FILE AND/OR DIRECTORY ENTRY
                     0070   633                  MODIFY,-                   ;MODIFY FILE ATTRIBUTES
                     0070   634                  MOUNT>                     ;MOUNT VOLUME
                     0078   635         FUNCTAB,-                           ;BUFFERED I/O FUNCTIONS
                     0078   636                 <NOP,-                      ;NO OPERATION
                     0078   637                  UNLOAD,-                   ;UNLOAD VOLUME
                     0078   638                  SPACERECORD,-              ;SPACE RECORDS
                     0078   639                  RECAL,-                    ;RECALIBRATE (REWIND)
                     0078   640                  DRVCLR,-                   ;DRIVE CLEAR
                     0078   641                  READPRESET,-               ;READ PRESET
                     0078   642                  PACKACK,-                  ;PACK ACKNOWLEDGE
                     0078   643                  ERASETAPE,-                ;ERASE TAPE
                     0078   644                  SENSECHAR,-                ;SENSE CHARACTERISTICS
                     0078   645                  SETCHAR,-                  ;SET CHARACTERISTICS
```

```
                                                ;SPACE FILES
                                                ;WRITE TAPE MARK
                                                ;**NEW**WRITE TAPE MARK RETRY
                                                ;**NEW**CLEAN TAPE
                                                ;SENSE MODE
                                                ;SET MODE
                                                ;REWIND
                                                ;REWIND AND UNLOAD
                                                ;SKIP RECORDS
                                                ;SKIP FILES
                                                ;WRITE END OF FILE
                                                ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
                                                ;ACP CONTROL FUNCTION
                                                ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
                                                ;DEACCESS FILE
                                                ;DELETE FILE AND/OR DIRECTORY ENTRY
                                                ;MODIFY FILE ATTRIBUTES
                                                ;MOUNT VOLUME
0080  664  FUNCTAB +ACP$READBLK,-               ;READ FUNCTIONS
0080  665         <READLBLK,-                   ;READ LOGICAL BLOCK FORWARD
0080  666          READPBLK,-                   ;READ PHYSICAL BLOCK FORWARD
0080  667          REREADN,-                    ;*NEW*REREAD NEXT
0080  668          REREADP,-                    ;*NEW*REREAD PREVIOUS
0080  669          READVBLK>                    ;READ VIRTUAL BLOCK
008C  670  FUNCTAB +ACP$WRITEBLK,-              ;WRITE FUNCTIONS
008C  671         <WRITECHECK,-                 ;WRITE CHECK FORWARD
008C  672          WRITELBLK,-                  ;WRITE LOGICAL BLOCK
008C  673          WRITEPBLK,-                  ;WRITE PHYSICAL BLOCK
008C  674          WRITERET,-                   ;*NEW*WRITE RETRY
008C  675          WRITEVBLK>                   ;WRITE VIRTUAL BLOCK
0098  676  FUNCTAB +ACP$ACCESS,<ACCESS,CREATE> ;ACCESS AND CREATE FILE OR DIRECTORY
00A4  677  FUNCTAB +ACP$DEACCESS,<DEACCESS>    ;DEACCESS FILE
00B0  678  FUNCTAB +ACP$MODIFY,-               ;
00B0  679         <ACPCONTROL,-                 ;ACP CONTROL FUNCTION
00B0  680          DELETE,-                     ;DELETE FILE OR DIRECTORY ENTRY
00B0  681          MODIFY>                      ;MODIFY FILE ATTRIBUTES
00BC  682  FUNCTAB +ACP$MOUNT,<MOUNT>           ;MOUNT VOLUME
00C8  683  FUNCTAB +MT$CHECK_ACCESS,-           ;MAGTAPE CHECK ACCESS FUNCITONS
00C8  684         <ERASETAPE,-                  ;ERASE TAPE
00C8  685          CLEAN,-                      ;**NEW**CLEAN TAPE
00C8  686          WRITEMARK,-                  ;WRITE TAPE MARK
00C8  687          WRTTMKR,-                    ;*NEW*WRITE TAPE MARK RETRY
00C8  688          WRITEOF>                     ;WRITE END OF FILE
00D4  689  FUNCTAB +EXE$ZEROPARM,-              ;ZERO PARAMETER FUNCTIONS
00D4  690         <NOP,-                        ;NO OPERATION
00D4  691          UNLOAD,-                     ;UNLOAD VOLUME
00D4  692          RECAL,-                      ;RECALIBRATE (REWIND)
00D4  693          REWIND,-                     ;REWIND
00D4  694          REWINDOFF,-                  ;REWIND AND SET OFFLINE
00D4  695          DRVCLR,-                     ;DRIVE CLEAR
00D4  696          READPRESET,-                 ;READ IN PRESET
00D4  697          PACKACK,-                    ;PACK ACKNOWLEDGE
00D4  698          ERASETAPE,-                  ;ERASE TAPE
00D4  699          CLEAN,-                      ;**NEW**CLEAN TAPE
00D4  700          SENSECHAR,-                  ;SENSE TAPE CHARACTERISTICS
00D4  701          SENSEMODE,-                  ;SENSE TAPE MODE
00D4  702          AVAILABLE,-                  ;AVAILABLE (REWIND/NOWAIT CLEAR VALID)
```

```
                                                B 4

00D4   703                      WRITEMARK,-            ;WRITE TAPE MARK
00D4   704                      WRTTMKR,-              ;*NEW*WRITE TAPE MARK RETRY
00D4   705                      WRITEOF>               ;WRITE END OF FILE
00E0   706              FUNCTAB +EXE$ONEPARM,-         ;ONE PARAMETER FUNCTIONS
00E0   707                      <SPACERECORD,-         ;SPACE RECORDS
00E0   708                       SPACEFILE,-           ;SPACE FILES
00E0   709                       SKIPRECORD,-          ;SKIP RECORDS
00E0   710                       SKIPFILE>             ;SKIP FILES
00EC   711              FUNCTAB +EXE$SETMODE,-         ;SET TAPE CHARACTERISTICS
00EC   712                      <SETCHAR,-             ;
00EC   713                      SETMODE>               ;
00F8   714
```

```
                                .IF     DF      TS_TRACE
                                .SBTTL  +       TRACE_IRP and TRACE_STATUS

                        ; Routines to record IRP and I/O status contents in the trace table.
                        ;       Trace table entries are 96 bytes long so that they line up nicely in
                        ;       a dump.
                        ;
                        ; TRACE_IRP
                        ;
                        ; Inputs:
                        ;       R3 => IRP
                        ;       R5 => UCB

                        TRACE_IRP:

                                BBC     #TRACE_V_ACTIVE,-       ; If trace table not intialized,
                                        UCB$W_TRACESTS(R5),20$  ;  branch around.
                                MOVQ    R0,-(SP)                ; Save R0 and R1.
                                MOVL    R3,R0                   ; R0 => IRP to trace.
                                CMPL    UCB$L_TRACEND(R5),-     ; See if we should circle back to start
                                        UCB$L_TRACEPTR(R5)      ;  of trace table.
                                BGTR    10$                     ; GTR implies NO.
                                MOVL    UCB$L_TRACEBEG(R5),-    ; TRACE_PTR => base of trace table.
                                        UCB$L_TRACEPTR(R5)
                        10$:
                                MOVL    UCB$L_TRACEPTR(R5),R1   ; R1 => area in trace table to use.

                                MOVQ    (R0)+,(R1)+             ; Twelve quad words are 96 bytes.
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+
                                MOVQ    (R0)+,(R1)+

                                MOVL    UCB$L_TRACEPTR(R5),R1   ; R1 => area in trace table to use.
                                MOVL    R3,(R1)                 ; Trace entry => IRP.
                                MNEGL   #1,4(R1)                ; Init flag field.
                                MNEGL   #1,IRP$L_ARB(R1)        ; Init field for I/O Status #1.
                                MNEGL   #1,IRP$L_ARB+4(R1)      ; Init field for I/O Status #2.
                                MOVQ    (SP)+,R0                ; Restore R0 and R1.
                        20$:
                                RSB

                        ; TRACE_STATUS
                        ;
                        ; Inputs:
                        ;
                        ;       R0 = I/O status value #1.
                        ;       R5 => UCB
                        ;       UCB$L_DEVDEPEND = I/O status #2.
```

```
00F8   773 TRACE_STATUS:
00F8   774
00F8   775          BBC     #TRACE_V_ACTIVE,-          ; If Trace table not active, branch.
00F8   776                  UCB$W_TRACESTS(R5),30$
00F8   777          PUSHL   R2                         ; Save register.
00F8   778          MOVL    UCB$L_TRACEPTR(R5),R2      ; R2 => area in trace table to use.
00F8   779
00F8   780          MOVL    R0,IRP$L_ARB(R2)           ; Save I/O status.
00F8   781          MOVL    UCB$L_DEVDEPEND(R5),-      ;    "
00F8   782                  IRP$L_ARB+4(R2)
00F8   783          POPL    R2                         ; Restore register.
00F8   784          ADDL    #96,UCB$L_TRACEPTR(R5)             ; Point to next entry.
00F8   785 30$:
00F8   786          RSB                                ; Return to caller.
00F8   787
00F8   788          .ENDC
```

```
                     00F8    790 .SBTTL  UNIT INITIALIZATION ROUTINE
                     00F8    791 ;+
                     00F8    792 ; THIS ROUTINE IS CALLED WHEN THE DRIVER IS LOADED OR ON POWERFAIL
                     00F8    793 ; RECOVERY.
                     00F8    794 ;
                     00F8    795 ; CALLING SEQUENCE:
                     00F8    796 ;       JSB     TS_INIT
                     00F8    797 ;
                     00F8    798 ; INPUT:
                     00F8    799 ;       R5 = UCB ADDRESS
                     00F8    800 ;       R4 = EQUIVALENT CSR FOR TS11
                     00F8    801 ;
                     00F8    802 ; OUTPUT:
                     00F8    803 ;
                     00F8    804
                     00F8    805 TS_INIT:
          10   A8    00F8    806         BISW    #UCB$M_ONLINE,-         ; Always mark TS11 as online since
        64 A5         00FA    807                 UCB$W_STS(R5)           ;  interrupts are not normally enabled
                      00FC    808                                         ;  we have no method to set it on
                      00FC    809                                         ;  dynamically.
             05   11  00FC    810         BRB     50$                     ; Go to allocate buffer and load registers
                      00FE    811 15$:
          10   AA    00FE    812         BICW    #UCB$M_ONLINE,-         ; Only reason for marking TS11 offline
        64 A5         0100    813                 UCB$W_STS(R5)           ;  is lack of pool space for PACKET.
                      0102    814 20$:
             05       0102    815         RSB                             ;RETURN
                      0103    816 50$:
                      0103    817         .IF     DF      TS_TRACE
                      0103    818         BBS     #TRACE_V_ACTIVE,-       ; If trace table already intialized,
                      0103    819                 UCB$W_TRACESTS(R5),52$  ;  branch around.
                      0103    820         MOVL    #50*96+16,R1            ; Allocate trace table for 50 entries.
                      0103    821         PUSHL   G^EXE$GL_NONPAGED               ; Save nonpaged IPL.
                      0103    822         MFPR    #PR$_IPL,G^EXE$GL_NONPAGED      ; Use current IPL.
                      0103    823         JSB     G^EXE$ALONONPAGED               ; Get from non-paged memory.
                      0103    824         POPL    G^EXE$GL_NONPAGED               ; Restore nonpaged IPL.
                      0103    825         BLBC    R0,52$                  ; Space not available, branch around.
                      0103    826
                      0103    827         CLRQ    (R2)+                   ; Initialize trace table header for SDA.
                      0103    828         MOVW    R1,(R2)+                ; Save size.
                      0103    829         MOVW    #DYN$C_SCS,(R2)+        ; Type.
                      0103    830         CLRL    (R2)+                   ; Round header upto 16 byte boundary.
                      0103    831         MOVL    R2,UCB$L_TRACEBEG(R5)           ; Save pointer to base of trace tabl
                      0103    832         MOVL    R2,UCB$L_TRACEPTR(R5)           ; Pointer to next area to use.
                      0103    833         ADDL3   #50*96,R2,-             ; Pointer to beyond end of trace
                      0103    834                 UCB$L_TRACEND(R5)       ;  table.
                      0103    835         BISW    #TRACE_M_ACTIVE,-       ; Indicate Trace table inited.
                      0103    836                 UCB$W_TRACESTS(R5)
                      0103    837 52$:
                      0103    838         .ENDC
   51   24   A5   D0  0103    839         MOVL    UCB$L_CRB(R5),R1        ;GET POINTER TO CRB
        34   A1   D0  0107    840         MOVL    CRB$L_INTD+VEC$W_MAPREG(R1),-
     00E8   C5        010A    841                 UCB$L_MS_TMP2(R5)       ;SAVE CURRENT UBA MAP CONTEXT.
   50   2C   A1   D0  010D    842         MOVL    CRB$L_INTD+VEC$L_IDB(R1),R0 ;GET POINTER TO IDB
   04   A0   55   D0  0111    843         MOVL    R5,IDB$L_OWNER(R0)      ;MAKE UCB OWNER OF IDB
        51   20   D0  0115    844         MOVL    #32,R1                  ;SIZE OF WORK BUFFER FOR TS11(=32.)
   52   00B6 C5   D0  0118    845         MOVL    UCB$L_MS_TSPT1(R5),R2   ;IF THE BUFFER HAS ALREADY BEEN ALLOCATED
             25   12  011D    846         BNEQ    60$                     ;BRANCH AROUND ELSE ALLOCATE THE BUFFER
```

```
                                                    011F    847
                                                    011F    848 ;
                                                    011F    849 ; DRIVER LOAD
                                                    011F    850 ;
                                                    011F    851
                                                    011F    852 55$:
        00000000'GF    DD                           011F    853          PUSHL   G^EXE$GL_NONPAGED        ;SAVE NONPAGED IPL
    00000000'GF    12    DB                         0125    854          MFPR    #PR$_IPL,G^EXE$GL_NONPAGED ;USE CURRENT IPL
        00000000'GF    16                           012C    855          JSB     G^EXE$ALONONPAGED       ;GET FROM NON-PAGED MEMORY
        00000000'GF  8ED0                           0132    856          POPL    G^EXE$GL_NONPAGED       ;RESTORE NONPAGED IPL
              03  50    E8                         0139    857          BLBS    R0,57$                  ; Space available, branch aroundd.
            FFBF    31                              013C    858          BRW     15$                     ; Branch on allocation failure.
                                                    013F    859 57$:
                                                    013F    860                                          ;YES, R1=SIZE OF ALLOCATED BLOCK
                                                    013F    861                                          ; R2 HAS ADDR. OF THE BLOCK
      00B6 C5    52    D0                         013F    862          MOVL    R2,UCB$L_MS_TSPT1(R5)   ;STORE ADDR. IN UCB
                                                    0144    863 60$:
                                                    0144    864          ASSUME  UCB$W_BOFF  EQ  UCB$L_SVAPTE+4
                                                    0144    865          ASSUME  UCB$W_BCNT  EQ  UCB$W_BOFF+2
            78  A5    7D                          0144    866          MOVQ    UCB$L_SVAPTE(R5),-
          00E0  C5                                  0147    867                  UCB$Q_MS_TMP1(R5)       ;SAVE UCB$L_SVAPTE, W_BCNT, W_BOFF.
                                                    014A    868
            7E  A5    51    B0                      014A    869          MOVW    R1,UCB$W_BCNT(R5)       ;LOAD BYTE COUNT INTO UCB
    7C  A5    52    FE00  8F    AB                  014E    870          BICW3   #^XFE00,R2,UCB$W_BOFF(R5) ;LOAD BYTE OFFSET IN UCB
        52    52    15    09    EF                  0155    871          EXTZV   S^#VA$V_VPN,S^#VA$S_VPN,R2,R2 ;GET VIRTUAL PAGE #
        50    00000000'GF       D0                  015A    872          MOVL    G^MMG$GL_SPTBASE,R0     ;GET ADDR. OF SYS. PAGE TABLE
            78  A5    6042      DE                  0161    873          MOVAL   (R0)[R2],UCB$L_SVAPTE(R5) ;STORE SVA OF PTE FOR WORK BUFFER
                                                    0166    874                                          ;LOADED BCNT,BOFF,&SVAPTE FOR WORK BUFFER
                                                    0166    875                                          ;DIRECT DATA PATH IS USED FOR COMMUNICATION
            51    24  A5    D0                      0166    876          MOVL    UCB$L_CRB(R5),R1        ; R1 => CRB
        34  A1      00D8  C5    D0                  016A    877          MOVL    UCB$L_MS_OMPR(R5),CRB$L_INTD+VEC$W_MAPREG(R1) ;IF NEQ USE OLD MAP RE
                        13    12                    0170    878          BNEQ    80$                     ;GOTO LOAD MAP REGISTER
                                                    0172    879 70$:
                    37  A1    94                    0172    880          CLRB    CRB$L_INTD+VEC$B_DATAPATH(R1) ;INSURE DIRECT DATA PATH(=0)
                                                    0175    881          REQMPR                          ;ALLOCATE MAP REGISTER(S) TO MAP UNIBUS
            51    24  A5    D0                      017B    882          MOVL    UCB$L_CRB(R5),R1        ;GET POINTER TO CRB
      00D8  C5    34  A1    D0                      017F    883          MOVL    CRB$L_INTD+VEC$W_MAPREG(R1),UCB$L_MS_OMPR(R5) ;SAVE OLD MAP REGISTER
                                                    0185    884 80$:
            78  A5    7D                          0185    885          MOVQ    UCB$L_SVAPTE(R5),-      ; Save message buffer parameters to
          00EC  C5                                  0188    886                  UCB$Q_MS_BUFSVAPTE(R5)  ;  facilitate later remapping.
                                                    018B    887                                          ;TO SBI ADDRESSES
                                                    018B    888                                          ; THE NO. OF MAP REGISTER AND STARTING
                                                    018B    889                                          ; MAP REG. NO. ARE STORED IN CRB
                                                    018B    890          LOADUBA                         ;LOAD MAP REG. TO BE USED
                                                    0191    891 ;
                                                    0191    892 ; CALCULATE UNIBUS ADDR. FOR COMMAND PACKET, STORE IT IN UCB
                                                    0191    893 ;
        50    7C  A5    3C                          0191    894          MOVZWL  UCB$W_BOFF(R5),R0       ;GET BYTE OFFSET
          00EC  C5    7D                          0195    895          MOVQ    UCB$Q_MS_TMP1(R5),-
            78  A5                                  0199    896                  UCB$L_SVAPTE(R5)        ;RESTORE SVAPTE, BOFF, BCNT
                                                    019B    897
            51    24  A5    D0                      019B    898          MOVL    UCB$L_CRB(R5),R1        ;GET CRB
50  09      09    34  A1    F0                      019F    899          INSV    CRB$L_INTD+VEC$W_MAPREG(R1),#9,#9,R0 ; HGM 9 BITS
        00BA  C5      50    D0                    01A5    900          MOVL    R0,UCB$L_MS_TSPT2(R5)   ;STORE IN UCB
          00E8  C5    D0                          01AA    901          MOVL    UCB$L_MS_TMP2(R5),-
            34  A1                                  01AE    902                  CRB$L_INTD+VEC$W_MAPREG(R1) ;RESTORE UNIBUS MAPPING CONTEXT
                51    D4                          01B0    903          CLRL    R1                      ;CLEAR R1
```

```
                50    50    FE  8F   78  01B2   904          ASHL    #-2,R0,R0                       ;MODULO 4,SHIFT OUT 2 0'S
          51    0E    02    50   F0  01B7   905          INSV    R0,#2,#14,R1                    ;INSERT B2-B15
                50    50    F2  8F   78  01BC   906          ASHL    #-14,R0,R0                      ;SHIFT OUT B2-B15
          51    02    00    50   F0  01C1   907          INSV    R0,#0,#2,R1                     ;INSERT B16-B17
                00BE  C5    51   B0  01C6   908          MOVW    R1,UCB$W_MS_TSPT3(R5)           ;STORE COMMAND PTR IN UCB
                                   01CB   909
                                   01CB   910 ;
                                   01CB   911 ; ISSUE WRITE CHARACTERISTIC COMMAND TO TELL MESSAGE BUFFER ADDR. TO TS11
                                   01CB   912 ;
                50    00B6  C5    D0  01CB   913          MOVL    UCB$L_MS_TSPT1(R5),R0           ;COMMAND PACKET ADDR. IN R0
                60    C084  8F    B0  01D0   914          MOVW    #<HC_DRC!MS_CPHD_M_ACK>,MS_CPHD(R0) ;GET COMMAND PACKET HEADER
          02    A0    00BA  C5    D0  01D5   915          MOVL    UCB$L_MS_TSPT2(R5),MS_BACT(R0)  ;STORE CHAR. BUFFER ADDR.
                02    A0    08  C0  01DB   916          ADDL    #8,MS_BACT(R0)                  ;POINT TO CHAR. BUFFER NOW
                06    A0    08  B0  01DF   917          MOVW    #8,MS_CNT(R0)                   ;STORE BYTE COUNT FOR CHAR. DATA
          08    A0    00BA  C5    D0  01E3   918          MOVL    UCB$L_MS_TSPT2(R5),MS_MBA0(R0)  ;STORE MESSAGE BUFFER ADDR.
                08    A0    10  C0  01E9   919          ADDL    #16,MS_MBA0(R0)                 ; AS CHAR. DATA
                0C    A0    0E  B0  01ED   920          MOVW    #14,MS_LNTH(R0)                 ;LENGTH OF CHAR. DATA=14.
                      0E    A0  B4  01F1   921          CLRW    MS_CHWD(R0)                     ;ZERO CHARACTERISTIC WORD
                                   01F4   922                                                  ;***=>NO MESSAGE BUFFER RELEASE INTERRUPT
                                   01F4   923                                                  ;**   ,NO ATTENTION INTERRUPT, AND NO
                                   01F4   924                                                  ;**  SKIP TAPE MARKS STOP
                                   01F4   925 ;
                                   01F4   926 ; LOAD COMMAND PTR IN DEVICE REGISTER TSDB, UCB
                                   01F4   927 ;
                64    00BE  C5    B0  01F4   928          MOVW    UCB$W_MS_TSPT3(R5),(R4) ;LOAD INTO TSDB
          68    A5    0400  8F    A8  01F9   929          BISW    #UCB$M_MS_LBA,UCB$W_DEVSTS(R5) ;MARK LOADING MESSAGE BUFFER
                                   01FF   930                                                  ; ADDR. INTO TS11.
                                05  01FF   931          RSB                                     ;RETURN
                                   0200   932
                                   0200   933
```

```
                                                H 4
                     0200   935          .SBTTL  TEST NBA (NEED BUFFER ADDRESS)
                     0200   936
                     0200   937 ; TEST_NBA - Subroutine called from STARTIO to determine if the TS11 has
                     0200   938 ;       a valid message buffer.  If YES, then we merely return.  If NOT, we
                     0200   939 ;       re-establish the message buffer obtained at SYSTEM INIT time.
                     0200   940 ;
                     0200   941 ;       This routine assumes that the following UCB fields were initialized
                     0200   942 ;       at UNIT INIT time:
                     0200   943 ;
                     0200   944 ;       UCB$Q_MS_BUFSVAPTE
                     0200   945 ;       UCB$L_MS_OMPR
                     0200   946 ;
                     0200   947 ; INPUTS:
                     0200   948 ;       R5 => UCB
                     0200   949 ;
                     0200   950 ; OUTPUTS:
                     0200   951 ;       Message buffer established in TS11.
                     0200   952 ;
                     0200   953
                     0200   954 TEST_NBA:
 009C C5 8ED0        0200   955          POPL    UCB$L_DPC(R5)           ; Pop return off stack in case.
 0400 8F   AA        0205   956          BICW    #UCB$M_MS_LBA,-         ; This bit maybe left on from INIT if
   68 A5             0209   957                  UCB$W_DEVSTS(R5)        ;  setting of switches inside drive so
                     020B   958                                          ;  dictate.  We clear it here because
                     020B   959                                          ;  this is a convenient place.
                     020B   960
51   24 A5   D0      020B   961          MOVL    UCB$L_CRB(R5),R1                ; R1 => CRB
                     020F   962          ASSUME  IDB$L_CSR  EQ  0
54   2C B1   D0      020F   963          MOVL    @CRB$L_INTD+VEC$L_IDB(R1),R4    ; R4 => CSR.
50   02 A4   B0      0213   964          MOVW    2(R4),R0                        ; R0 contains TSSR register.
03 50    07   E0      0217   965          BBS     #MS_TSSR_V_SSR,R0,10$   ; Branch to continue if TS11 ready.
       009A   31      021B   966          BRW     60$                     ; Branch to failure if NOT ready.
                     021E   967 10$:
07 50    0A   E0      021E   968          BBS     #MS_TSSR_V_NBA,R0,20$   ; Branch around if we NEED to re-
                     0222   969                                          ;  establish message buffer address.
   50    01   9A      0222   970          MOVZBL  S^#SS$_NORMAL,R0        ; Else indicate success and
 009C D5   17        0225   971          JMP     @UCB$L_DPC(R5)          ;  return to caller.
                     0229   972 20$:
   34 A1   D0        0229   973          MOVL    CRB$L_INTD+VEC$W_MAPREG(R1),-
 00E8 C5             022C   974                  UCB$L_MS_TMP2(R5)       ;SAVE CURRENT UBA MAP CONTEXT.
 00D8 C5   D0        022F   975          MOVL    UCB$L_MS_OMPR(R5),-     ; Setup to map UNIBUS just in case
   34 A1             0233   976                  CRB$L_INTD+VEC$W_MAPREG(R1)
                     0235   977
                     0235   978          ASSUME  UCB$W_BOFF  EQ  UCB$L_SVAPTE+4
                     0235   979          ASSUME  UCB$W_BCNT  EQ  UCB$W_BOFF+2
   78 A5   7D        0235   980          MOVQ    UCB$L_SVAPTE(R5),-
 00E0 C5             0238   981                  UCB$Q_MS_TMP1(R5)       ;SAVE UCB$L_SVAPTE, W_BCNT, W_BOFF.
 00EC C5   7D        023B   982          MOVQ    UCB$Q_MS_BUFSVAPTE(R5),-; Restore parameters to remap message
   78 A5             023F   983                  UCB$L_SVAPTE(R5)        ;  buffer in UNIBUS space.
                     0241   984
                     0241   985          LOADUBA                         ; Reload UNIBUS map registers for
                     0247   986                                          ;  message buffer.
                     0247   987
                     0247   988 ;
                     0247   989 ; ISSUE WRITE CHARACTERISTIC COMMAND TO TELL MESSAGE BUFFER ADDR. TO TS11
                     0247   990 ;
50   00B6 C5   D0      0247   991          MOVL    UCB$L_MS_TSPT1(R5),R0   ; R0 => command packet
```

```
             C084 8F   B0 024C        MOVW    #<HC_WRC!MS_CPHD_M_ACK>,-
                  60      0250                MS_CPHD(R0)             ; Move command (WRITE CHARACTERISTICS)
                          0251                                        ;  to 1ST word of command packet.
             00BA C5   D0 0251        MOVL    UCB$L_MS_TSPT2(R5),-    ; Store UNIBUS address of packet in
               02 A0      0255                MS_BACT(R0)             ;  packet.
          02 A0   08   C0 0257        ADDL    #8,MS_BACT(R0)          ; Update to point to CHARACTERISTICS
                          0258                                        ;  buffer beyond packet.
          06 A0   08   B0 025B        MOVW    #8,MS_CNT(R0)           ;Store byte count for char. data
             00BA C5   D0 025F        MOVL    UCB$L_MS_TSPT2(R5),-    ; Store UNIBUS address of PACKET
               08 A0      0263                MS_MBAO(R0)             ;  into the CHARACTERISTICS data.
          08 A0   10   C0 0265        ADDL    #16,MS_MBAO(R0)         ; Message BUFF is 16 beyond packet.
                          0269
          0C A0   0E   B0 0269        MOVW    #14,MS_LNTH(R0)         ;LENGTH OF CHAR. DATA=14.
               0E A0   B4 026D        CLRW    MS_CHWD(R0)             ;ZERO CHARACTERISTIC WORD
                          0270                               ;**=>NO MESSAGE BUFFER RELEASE INTERRUPT
                          0270                               ;**  ,NO ATTENTION INTERRUPT, AND NO
                          0270                               ;**  SKIP TAPE MARKS STOP
                          0270        DSBINT
       17 64 A5   05   E0 0276        BBS     #UCB$V_POWER,UCB$W_STS(R5),30$
       64    00BE C5   B0 027B        MOVW    UCB$W_MS_TSPT3(R5),(R4) ;LOAD INTO TSDB
                          0280        WFIKPCH 40$,#2                  ; Wait for interrupt.
                          028A        IOFORK
                  09   11 0290        BRB     50$                     ; Branch around powerfail branch.
                          0292  30$:  ENBINT
                          0295  40$:  SETIPL  UCB$B_FIPL(R5)          ; Lower IPL in case of TIMEOUT.
                  1D   11 0299        BRB     60$                     ; Branch if we had POWERFAIL.
                          029B
                          029B  50$:
       51    24 A5     D0 029B        MOVL    UCB$L_CRB(R5),R1        ; R1 => CRB.
             00E8 C5   D0 029F        MOVL    UCB$L_MS_TMP2(R5),-     ; Restore previous mapping context.
               34 A1      02A3                CRB$L_INTD+VEC$W_MAPREG(R1)
             00E0 C5   7D 02A5        MOVQ    UCB$Q_MS_TMP1(R5),-     ; And also transfer parameters.
               78 A5      02A9                UCB$L_SVAPTE(R5)
                  0A   E0 02AB        BBS     #MS_TSSR_V_NBA,-        ; Test if all that had any effect
          07 00C2 C5      02AD                UCB$W_MS_TSSR(R5),60$   ;  by seeing if we still have NBA.
          50      01   9A 02B1        MOVZBL  S^#SS$_NORMAL,R0        ; Else indicate success and
             009C D5   17 02B4        JMP     @UCB$L_DPC(R5)          ;  return to caller.
                          02B8  60$:
                          02B8                                        ;  Terminate the I/O function
       50    0084 8F   3C 02B8        MOVZWL  #SS$_DEVOFFLINE,R0      ;  by returning the OFFLINE status and
             009C D5   17 02BD        JMP     @UCB$L_DPC(R5)          ;  return to caller.
```

```
                        02C1  1034 .SBTTL  START I/O OPERATION
                        02C1  1035 ;+
                        02C1  1036 ; TS_STARTIO - START I/O OPERATION ON DEVICE
                        02C1  1037 ;
                        02C1  1038 ; THIS ENTRY POINT IS ENTERED TO START AN I/O OPERATION ON TS11/TS04
                        02C1  1039 ;
                        02C1  1040 ; INPUTS:
                        02C1  1041 ;
                        02C1  1042 ;       R3 = ADDRESS OF I/O PACKET.
                        02C1  1043 ;       R5 = UCB ADDRESS OF DEVICE UNIT
                        02C1  1044 ;
                        02C1  1045 ; OUTPUT:
                        02C1  1046 ;
                        02C1  1047 ;       FUNCTION DEPENDENT PARAMETERS ARE STORED INTO THE DEVICE UCB,
                        02C1  1048 ;       ERROR RETRY COUNT IS RESET, AND THE FUNCTION IS EXECUTED.  AT
                        02C1  1049 ;       FUNCTION COMPLETION THE OPERATION IS TERMINATED THRU REQUEST COMPLETE.
                        02C1  1050 ;
                        02C1  1051 ;-
                        02C1  1052
                        02C1  1053 TS_STARTIO:                                  ;START I/O OPERATION
                        02C1  1054         .IF     DF      TS_TRACE
                        02C1  1055         BSBW    TRACE_IRP                    ; Trace this IRP.
                        02C1  1056         .ENDC
                FF3C 30 02C1  1057         BSBW    TEST_NBA                     ; Assure that TS11 has valid MESSAGE
                        02C4  1058                                              ;  BUFFER.
              03 50 E8  02C4  1059         BLBS    R0,5$                        ; LBS implies success. GOTO continue.
                04CB 31 02C7  1060         BRW     FCNEXT                       ;  Else branch to terminate function.
                        02CA  1061 5$:
0080 C5  0081 C5 90     02CA  1062         MOVB    UCB$B_ERTMAX(R5),UCB$B_ERTCNT(R5) ;INITIALIZE ERROR RETRY COUNT
 009A C5   20 A3 B0     02D1  1063         MOVW    IRP$W_FUNC(R3),UCB$W_FUNC(R5) ;SAVE FUNCTION CODE & MODIFIER
       50   38 A3 D0     02D7  1064         MOVL    IRP$L_MEDIA(R3),R0           ;GET PARAMETER LONGWORD
                        02DB  1065
                        02DB  1066 ;
                        02DB  1067 ; MOVE FUNCTION DEPENDENT PARAMETERS TO UCB
                        02DB  1068 ;
                        02DB  1069
    06    00    EF      02DB  1070         EXTZV   #IRP$V_FCODE,#IRP$S_FCODE,- ;EXTRACT I/O FUNCTION CODEE
51     20 A3            02DE  1071                 IRP$W_FUNC(R3),R1            ;
    51    02    D1      02E1  1072         CMPL    #IO$_SPACEFILE,R1            ;SPACE FILE FUNCTION?
          2B    13      02E4  1073         BEQL    10$                          ;IF EQL YES
    51    09    D1      02E6  1074         CMPL    #IO$_SPACERECORD,R1          ;SPACE RECORD FUNCTION?
          32    13      02E9  1075         BEQL    20$                          ;IF EQL YES
    51    1A    D1      02EB  1076         CMPL    #IO$_SETCHAR,R1              ;SET CHARACTERISTICS FUNCTION?
          46    13      02EE  1077         BEQL    50$                          ;IF EQL YES
    51    11    D1      02F0  1078         CMPL    #IO$_AVAILABLE,R1            ;AVAILABLE function?
          5A    13      02F3  1079         BEQL    75$                          ;IF EQL YES
    51    0D    D1      02F5  1080         CMPL    #IO$_READPBLK+1,R1           ;DISJOINT CODE?
          6C    1A      02F8  1081         BGTRU   100$                         ;IF GTRU NO
                        02FA  1082         CASE    R1,<-                        ;DISPATCH LOGICAL FUNCTIONS
                        02FA  1083                 70$,-                        ;REWIND AND SET OFFLINE
                        02FA  1084                 60$,-                        ;SET MODE
                        02FA  1085                 80$,-                        ;REWIND
                        02FA  1086                 10$,-                        ;SKIP FILE
                        02FA  1087                 20$,-                        ;SKIP RECORD
                        02FA  1088                 90$,-                        ;SENSE TAPE MODE
                        02FA  1089                 90$,-                        ;WRITE EOF
                        02FA  1090                 >,LIMIT=#IO$_REWINDOFF       ;
```

```
                     51   06   A2  030C  1091          SUBW    #IO$_READPRESET-IO$_READPBLK-7,R1 ;CONVERT TO DENSE FUNCTION CODE
                          63   11  030F  1092          BRB     110$                              ;**LAST LINE NEED BE ADJUSTED
                                   0311  1093
                                   0311  1094 ;
                                   0311  1095 ; SPACE FILE FUNCTION - SET SPACE COUNT AND PROPER FUNCTION
                                   0311  1096 ;
                                   0311  1097
                     51   02   3C  0311  1098 10$:     MOVZWL  #CDHC_STF,R1              ;SET SPACE FILE FORWARD
                          50   B5  0314  1099          TSTW    R0                        ;SPACE FILE FORWARD?
                          12   14  0316  1100          BGTR    40$                       ;IF GTR YES
                     51   05   9A  0318  1101          MOVZBL  #CDHC_STR,R1              ;SET FOR SPACE FILE REVERSE
                          0A   11  031B  1102          BRB     30$                       ;
                                   031D  1103
                                   031D  1104 ;
                                   031D  1105 ; SPACE RECORD FUNCTION - SET SPACE COUNT AND PROPER HARDWARE COMMAND
                                   031D  1106 ;
                                   031D  1107
                     51   09   9A  031D  1108 20$:     MOVZBL  #CDHC_SRF,R1              ;SET FOR SPACE RECORD FORWARD
                          50   B5  0320  1109          TSTW    R0                        ;SPACE RECORD FORWARD?
                          06   14  0322  1110          BGTR    40$                       ;IF GTR YES
                     51   07   9A  0324  1111          MOVZBL  #CDHC_SRR,R1              ;SET FOR SPACE RECORD REVERSE
                     50   50   AE  0327  1112 30$:     MNEGW   R0,R0                     ;CONVERT TO POSITIVE COUNT
              00B4   C5   50   B0  032A  1113 40$:     MOVW    R0,UCB$W_MS_SPACNT(R5)    ;SET SPACE COUNT
                          43   12  032F  1114          BNEQ    110$                      ;IF NEQ SPACING REQUIRED
                     51   00   9A  0331  1115          MOVZBL  #CDHC_NOP,R1              ;SET FOR NO OPERATION
                          3E   11  0334  1116          BRB     110$                      ;
                                   0336  1117
```

```
                     0336  1119 ;
                     0336  1120 ; SET CHARACTERISTICS FUNCTION - STORE NEW TAPE CHARACTERISTICS
                     0336  1121 ;
                     0336  1122 ;****TS11/TS04 HAS ONLY ONE CLASS AND TYPE****
                     0336  1123
40 A5   38 A3   B0   0336  1124 50$:    MOVW    IRP$L_MEDIA(R3),UCB$B_DEVCLASS(R5) ;SET NEW DEVICE CLASS AND TYPE
                     033B  1125
                     033B  1126 ;
                     033B  1127 ; SET MODE FUNCTION - STORE NEW TAPE MODE
                     033B  1128 ;
                     033B  1129
42 A5   3A A3   B0   033B  1130 60$:    MOVW    IRP$L_MEDIA+2(R3),UCB$W_DEVBUFSIZ(R5) ;SET NEW DEFAULT BUFFER SIZE
7C A5   3C A3   B0   0340  1131         MOVW    IRP$L_MEDIA+4(R3),UCB$W_BOFF(R5) ;SAVE NEW TAPE CONTROL PARAMETERS
        51      14   9A   0345  1132         MOVZBL  #CDHC_SCH,R1            ;SET DISPATCH INDEX
                2A   11   0348  1133         BRB     110$                    ;
                     034A  1134
                     034A  1135 ;
                     034A  1136 ; LOGICAL REWIND AND SET TAPE OFFLINE - CONVERT TO UNLOAD COMMAND
                     034A  1137 ;
                     034A  1138
        51      01   9A   034A  1139 70$:    MOVZBL  #CDHC_UNL,R1            ;SET FOR UNLOAD COMMAND
                25   11   034D  1140         BRB     110$                    ;
                     034F  1141
                     034F  1142 ;
                     034F  1143 ; AVAILABLE FUNCTION - Equivalent of REWIND(NOWAIT) and clear of UCB$M_VALID.
                     034F  1144 ;
                     034F  1145
                     034F  1146 75$:
        00A4 8F      B0   034F  1147         MOVW    #IO$_REWIND!IO$M_NOWAIT,-; Simulate a REWIND NOWAIT.
             009A C5      0353  1148                 UCB$W_FUNC(R5)
        0800 8F      AA   0356  1149         BICW    #UCB$M_VALID,-          ; And clear valid bit.
             64 A5        035A  1150                 UCB$W_STS(R5)           ;  and fall thru to rewind logic.
                          035C  1151
                          035C  1152 ;
                          035C  1153 ; LOGICAL REWIND FUNCTION - CONVERT TO PHYSICAL FUNCTION
                          035C  1154 ;
                          035C  1155
        51      03   9A   035C  1156 80$:    MOVZBL  #CDHC_RWD,R1            ;SET FOR REWIND
                13   11   035F  1157         BRB     110$                    ;
                          0361  1158
                          0361  1159 ;
                          0361  1160 ; LOGICAL WRITE EOF OR SENSE MODE FUNCTION - CONVERT TO PHYSICAL FUNCTION
                          0361  1161 ;
                          0361  1162
        51      12   A2   0361  1163 90$:    SUBW    #IO$_SENSEMODE-IO$_READPBLK-9,R1 ;CONVERT TO PHYSICAL****
                0E   11   0364  1164         BRB     110$                    ;
                          0366  1165
                          0366  1166 ;
                          0366  1167 ; DENSE FUNCTION CODE - CHECK FOR READ, WRITE, OR WRITECHECK FUNCTION
                          0366  1168 ;
                          0366  1169
        51      0A   D1   0366  1170 100$:   CMPL    #IO$_WRITECHECK,R1       ;DATA TRANSFER FUNCTION?
                09   1A   0369  1171         BGTRU   110$                     ;IF GTRU NO
03 009A C5      06   E1   036B  1172         BBC     #IO$V_REVERSE,UCB$W_FUNC(R5),110$ ;IF CLEAR,NOT REVERSE
        51      03   A0   0371  1173         ADDW    #CDHC_WKR-CDHC_WCK,R1    ;CONVERT TO REVERSE FUNCTION
                          0374  1174
                          0374  1175 ;
```

```
                        0374  1176 ; FINISH PREPROCESSING
                        0374  1177 ;
                        0374  1178
0092 C5   51    90      0374  1179 110$:   MOVB    R1,UCB$B_FEX(R5)        ;SAVE FUNCTION DISPATCH INDEX
                        0379  1180                                        ;****NOTE ABOUT BYTE FOR INDEX
                        0379  1181
                        0379  1182 ;
                        0379  1183 ;CENTRAL FUNCTION DISPATCH
                        0379  1184 ;
                        0379  1185
                        0379  1186 FDISPATCH:
   53    58 A5    D0      0379  1187         MOVL    UCB$L_IRP(R5),R3        ;RETRIEVE ADDR. OF I/O PACKET
0D 2A A3    08    E0      037D  1188         BBS     #IRP$V_PHYSIO,IRP$W_STS(R3),10$ ;IF SET, PHYSICAL I/O FUNCTION
08 64 A5    0B    E0      0382  1189         BBS     #UCB$V_VALID,UCB$W_STS(R5),10$ ;IF SET, VOLUME SOFTWARE VALID
 50    0254 8F    3C      0387  1190         MOVZWL  #SS$_VOLINV,R0          ;SET VOLUME INVALID STATUS
          0406    31      038C  1191         BRW     FCNEXT          ;**NO CHANGE ON UCB$V_VALID BIT HERE**
                        038F  1192
                        038F  1193 ;
                        038F  1194 ; UNIT IS SOFTWARE VALID OR FUNCTION IS PHYSICAL I/O
                        038F  1195 ;
                        038F  1196
                        038F  1197 10$:
   54    24 A5    D0      038F  1198         MOVL    UCB$L_CRB(R5),R4        ;GET CSR ADDRESS INTO R4...
   54    2C B4    D0      0393  1199         MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4 ;
 50    0092 C5    9A      0397  1200         MOVZBL  UCB$B_FEX(R5),R0        ;GET DISPATCH INDEX
                        039C  1201         CASE    R0,<-                   ;DISPATCH TO COMMAND HANDLING ROUTINE
                        039C  1202                 NOP,-                   ;NO OPERATION
                        039C  1203                 UNLOAD,-                ;REWIND & UNLOAD
                        039C  1204                 SPCFILFOR,-             ;SPACE FILE FORWARD
                        039C  1205                 REWIND,-                ;REWIND
                        039C  1206                 DRVCLR,-                ;DRIVE CLEAR
                        039C  1207                 SPCFILREV,-             ;SPACE FILE REVERSE
                        039C  1208                 ERASE,-                 ;ERASE
                        039C  1209                 SPCRECREV,-             ;SPACE RECORD REVERSE
                        039C  1210                 PACKACK,-               ;PACK ACKNOWLEDGE
                        039C  1211                 SPCRECFOR,-             ;SPACE RECORD FORWARD
                        039C  1212                 WRITECHECK,-            ;SIMULATED WRITECHECK
                        039C  1213                 WRITEDATA,-             ;WRITE DATA FORWARD
                        039C  1214                 READDATA,-              ;READ DATA FORWARD
                        039C  1215                 WRITECHECKR,-           ;WRITE CHECK REVERSE
                        039C  1216                 WRITEDATA,-             ;WRITE DATA(NO REVERSE)
                        039C  1217                 READDATAR,-             ;READ DATA REVERSE
                        039C  1218                 REREADN,-               ;REREAD DATA NEXT
                        039C  1219                 REREADP,-               ;REREAD DATA PREVIOUS
                        039C  1220                 WRITERET,-              ;WRITE DATA RETRY
                        039C  1221                 READPRESET,-            ;SIMULATED READ PRESET
                        039C  1222                 SETCHAR,-               ;SIMULATED SET CHARACTERISTIC
                        039C  1223                 GETSTS,-                ;GET STATUS IMMEDIATE(SENS CHAR.)
                        039C  1224                 WRTTMK,-                ;WRITE TAPE MARK
                        039C  1225                 WRTTMKR,-               ;WRITE TAPE MARK RETRY
                        039C  1226                 CLEAN,-                 ;CLEAN
                        039C  1227                 MSGREL,-                ;MESSAGE BUFFER RELEASE
                        039C  1228                 WRITESUBS,-             ;WRITE SUBSYSTEM MEMEORY
                        039C  1229                 WRITECHAR,-             ;WRITE CHARACTERISTICS
                        039C  1230                 >
                        03D8  1231                                         ;***NOTE INDEX OUT OF BOUND***
```

```
                        03D8  1233 .SBTTL  NOP AND SIMULATED FUNCTIONS
                        03D8  1234
                        03D8  1235 ;
                        03D8  1236 ; SET CHARACTERISTIC FUNCTION
                        03D8  1237 ;
                        03D8  1238
                        03D8  1239 SETCHAR:                                             ;SET CHARACTERISTIC
    68 A5   02   AA     03D8  1240          BICW    #UCB$M_MS_SWAP,UCB$W_DEVSTS(R5) ;CLEAR SWAP BIT 1ST
                04   EF 03DC  1241          EXTZV   #MT$V_FORMAT,-                  ;GET FORMAT FIELD
                04      03DE  1242                  #MT$S_FORMAT,-                  ;
    51   7C A5          03DF  1243                  UCB$W_BOFF(R5),R1               ;
         51     0E   B1 03E2  1244          CMPW    #MT$K_NORMAL15,R1               ;IS IT INDUSTRIAL COMPATIBLE?
                04   12 03E5  1245          BNEQ    5$                              ;BR IF NO
    68 A5       02   A8 03E7  1246          BISW    #UCB$M_MS_SWAP,UCB$W_DEVSTS(R5) ;SET SWAP BIT FOR
                        03EB  1247                                                  ; SUBSEQUENT IO FUNCTIONS
                        03EB  1248 5$:
                        03EB  1249
                        03EB  1250 ;
                        03EB  1251 ; NO OPERATION AND SIMULATED NON-EXISTENT TS11/TS04 HARDWARE COMMAND
                        03EB  1252 ;
                        03EB  1253
                        03EB  1254 PACKACK:                                             ;PACK ACKNOWLEDGE
  64 A5   0800 8F   A8  03EB  1255          BISW    #UCB$M_VALID,UCB$W_STS(R5)      ; PACKACK implies set volume valid.
                        03F1  1256 NOP:                                                 ;NO OPERATION
                        03F1  1257 WRITECHECK:                                          ;SIMULATED WRITECHECK
                        03F1  1258 WRITECHECKR:                                         ;SIMULATED WRITECHECK REVERSE
                        03F1  1259 READPRESET:                                          ;READ IN PRESET
                        03F1  1260
                        03F1  1261          EXHC    10$                             ;EXECUTE HARDWARE COMMAND, IF ANY
                        03F6  1262 10$:
 039C   31              03F6  1263          BRW     FCNEXT                          ;GOTO FUNCTION EXIT
                        03F9  1264                                                  ;10$ AS RETRIABLE ERROR OCCURRED
                        03F9  1265                                      ;NO RETRIABLE ERROR, NOP ALWAYS SUCCESSFUL
                        03F9  1266
```

```
                     03F9  1268 .SBTTL  READ HARDWARE FUNCTIONS
                     03F9  1269
                     03F9  1270 ;
                     03F9  1271 ; READ HARDWARE FUNCTIONS
                     03F9  1272 ;
                     03F9  1273
                     03F9  1274 READDATA:                                    ;READ DATA FORWARD
                     03F9  1275         EXHC    10$                          ;EXECUTE HARDWARE COMMAND
      00B0 C5   D6   03FE  1276         INCL    UCB$L_RECORD(R5)             ;INCREMENT RECORD COUNT
         0390   31   0402  1277         BRW     FCNEXT                       ;GOTO SUCCESSFUL RETURN
                     0405  1278                                 ;10$ HANDLES RETRIABLE ERRORS
                     0405  1279 10$:
            50   DD   0405  1280         PUSHL   R0                           ;SAVE R0 HAS TCC
00000000'GF   16   0407  1281         JSB     G^ERL$DEVICERR               ;LOG BEFORE RETRY
            50 8ED0   040D  1282         POPL    R0                           ;RESTORE
                     0410  1283 20$:
      50   04   D1   0410  1284         CMPL    #TCC_REM,R0                  ;DID TAPE MOVED
            15   13   0413  1285         BEQL    22$                          ;YES BRANCH
      0080 C5   97   0415  1286         DECB    UCB$B_ERTCNT(R5)                     ;ANY RETRIES REMAINING?
            24   19   0419  1287         BLSS    30$                 ;NO, GO AS FATAL
                     041B  1288         EXHC    20$,HC_RDN                   ;DO READ AGAIN
      00B0 C5   D6   0423  1289         INCL    UCB$L_RECORD(R5)             ;INCREMENT RECORD COUNNT
         036B   31   0427  1290         BRW     RFCNEXT                      ;SUCEED, RETURN
                     042A  1291 22$:
      0080 C5   97   042A  1292         DECB    UCB$B_ERTCNT(R5)                     ;ANY RETRIES REMAINING?
            0F   19   042E  1293         BLSS    30$                 ;NO, GO AS FATAL ERROR
                     0430  1294         EXHC    22$,HC_RRP                   ;DO REREAD PREVIOUS
      00B0 C5   D6   0438  1295         INCL    UCB$L_RECORD(R5)             ;INCREMENT RECORD COUNT
         0356   31   043C  1296         BRW     RFCNEXT                      ;SUCCEED, RETURN
                     043F  1297 30$:
0000077A'EF   17   043F  1298         JMP     FATALERO
                     0445  1299
                     0445  1300 ;
                     0445  1301 ; REREAD PREVIOUS (SPACE REV,READ FWD)
                     0445  1302 ;
                     0445  1303
                     0445  1304 REREADP:                                     ;REREAD DATA PREVIOUS
                     0445  1305         EXHC    10$                          ;
         0348   31   044A  1306         BRW     FCNEXT                       ;SUCCESS RETURN
                     044D  1307 10$:
         032A   31   044D  1308         BRW     FATALERO                     ;TREATED AS FATAL AS NOW
                     0450  1309
                     0450  1310 ;
                     0450  1311 ; READ PREVIOUS
                     0450  1312 ;
                     0450  1313
                     0450  1314 READDATAR:                                   ;READ DATA REVERSE
                     0450  1315         EXHC    10$
      00B0 C5   D7   0455  1316         DECL    UCB$L_RECORD(R5)             ;DECREMENT RECORD COUNT
                     0459  1317                                 ;*NOTE*TMK PROBLEM???
         0339   31   0459  1318         BRW     FCNEXT                       ;DO SUCCESSFUL RETURN
                     045C  1319 10$:                                         ;RETRIABLE
            50   DD   045C  1320         PUSHL   R0                           ;SAVE R0 WHICH HAS TCC CODE
00000000'GF   16   045E  1321         JSB     G^ERL$DEVICERR               ;LOG BEFORE RETRY
            50 8ED0   0464  1322         POPL    R0                           ;RESTORE
                     0467  1323 20$:
      50   04   D1   0467  1324         CMPL    #TCC_REM,R0                  ;TAPE MOVED?
```

```
                     046A  1325          BEQL    22$                     ;YES
0080 C5    97        046C  1326          DECB    UCB$B_ERTCNT(R5)        ;ANY RETRIES LEFT?
     24    19        0470  1327          BLSS    30$                     ;NO, AS FATAL ERROR
                     0472  1328          EXHC    20$,HC_RDP              ;DO READ DATA PREVIOUS AGAIN
00B0 C5    D7        047A  1329          DECL    UCB$L_RECORD(R5)        ;DECREMENT RECORD COUNT
   0314    31        047E  1330          BRW     RFCNEXT                 ;SUCCESS RETURN
                     0481  1331 22$:
0080 C5    97        0481  1332          DECB    UCB$B_ERTCNT(R5)        ;ANY RETRIES LEFT?
     0F    19        0485  1333          BLSS    30$                     ;NO, AS FATAL ERROR
                     0487  1334          EXHC    22$,HC_RRN              ;DO REREAD DATA NEXT
00B0 C5    D7        048F  1335          DECL    UCB$L_RECORD(R5)        ;DECREMENT RECORD COUNT
   02FF    31        0493  1336          BRW     RFCNEXT                 ;SUCCESS RETURN
                     0496  1337 30$:
   02E1    31        0496  1338          BRW     FATALERO
                     0499  1339
                     0499  1340
                     0499  1341 ;
                     0499  1342 ; REREAD DATA NEXT(SPACE FWD, READ REV)
                     0499  1343 ;
                     0499  1344
                     0499  1345 REREADN:                                 ;REREAD DATA NEXT
                     0499  1346          EXHC    10$
   02F4    31        049E  1347          BRW     FCNEXT                  ;SUCCESS RETURN
                     04A1  1348 10$:
   02D6    31        04A1  1349          BRW     FATALERO                ;AS FATAL ERROR AS NOW
                     04A4  1350
                     04A4  1351
```

```
                     04A4   1353 .SBTTL  WRITE FUNCTIONS
                     04A4   1354
                     04A4   1355 ;
                     04A4   1356 ; WRITE DATA
                     04A4   1357 ;
                     04A4   1358
                     04A4   1359 WRITEDATA:                                  ;WRITE DATA FORWARD
 46 A5   08    AA    04A4   1360         BICW    #<MT$M_HWL>@-16,UCB$L_DEVDEPEND+2(R5) ;CLEAR
                     04A8   1361                                             ; HARDWARE WRITE LOCK BIT
                     04A8   1362         EXHC    10$                         ;
   00B0 C5   D6      04AD   1363         INCL    UCB$L_RECORD(R5)            ;INCREMENT RECORD COUNT
      02E1   31      04B1   1364         BRW     FCNEXT                      ;TAKE FUNCTION EXIT
                     04B4   1365 10$:
        50   DD      04B4   1366         PUSHL   R0                          ;SAVE R0
00000000'GF   16      04B6   1367         JSB     G^ERL$DEVICERR              ;LOG BEFORE RETRY
        50 8ED0       04BC   1368         POPL    R0                          ;RESTORE
                     04BF   1369 20$:
    50   04   D1      04BF   1370         CMPL    #TCC_REM,R0                 ;TAPE MOVED?
        15   13      04C2   1371         BEQL    22$                         ;YES
   0080 C5   97      04C4   1372         DECB    UCB$B_ERTCNT(R5)            ;ANY RETRIES LEFT?
        24   19      04C8   1373         BLSS    30$                         ;NO, AS FATAL ERROR
                     04CA   1374         EXHC    20$,HC_WRD                  ;YES, DO WRITE AGAIN
   00B0 C5   D6      04D2   1375         INCL    UCB$L_RECORD(R5)            ;INCREMENT RECORD COUNT
      02BC   31      04D6   1376         BRW     RFCNEXT                     ;TAKE SUCCESS RETURN
                     04D9   1377 22$:
   0080 C5   97      04D9   1378         DECB    UCB$B_ERTCNT(R5)            ;ANY RETRIES LEFT?
        0F   19      04DD   1379         BLSS    30$                 ;NO, FATAL
                     04DF   1380         EXHC    22$,HC_WDR                  ;DO WRITE DATA RETRY
   00B0 C5   D6      04E7   1381         INCL    UCB$L_RECORD(R5)            ;INCREMENT RECORD COUNT
      02A7   31      04EB   1382         BRW     RFCNEXT                     ;SUCCESS RETURN
                     04EE   1383 30$:
      0289   31      04EE   1384         BRW     FATALERO
                     04F1   1385
                     04F1   1386 ;
                     04F1   1387 ; WRITE DATA RETRY(SPACE REV,ERASE,WRITE DATA)
                     04F1   1388 ;
                     04F1   1389
                     04F1   1390 WRITERET:                                   ;WRITE DATA RETRY
                     04F1   1391         EXHC    10$                         ;
   029C   31         04F6   1392         BRW     FCNEXT                      ;TAKE SUCCESS RETURN
                     04F9   1393 10$:
   027E   31         04F9   1394         BRW     FATALERO                    ;AS FATAL
                     04FC   1395
                     04FC   1396 ;
                     04FC   1397 ; WRITE SUBSYSTEM MEMORY
                     04FC   1398 ;
                     04FC   1399
                     04FC   1400 WRITESUBS:                                  ;WRITE SUBSYSTEM MEMORY
                     04FC   1401         EXHC    10$                         ;
   0291   31         0501   1402         BRW     FCNEXT                      ;SUCCESS RETURN
                     0504   1403 10$:
                     0504   1404
   0273   31         0504   1405         BRW     FATALERO
                     0507   1406
                     0507   1407 ;
                     0507   1408 ; WRITE CHARACTERISTICS
                     0507   1409 ; USED TO TELL SUBSYSTEM MSG BUFFER ADDR. & SET CHARACTERISTIC WORD
```

```
                     0507  1410 ;
                     0507  1411
                     0507  1412 WRITECHAR:                                ;WRITE CHARACTERISTICS
                     0507  1413
                     0507  1414          EXHC    10$                     ;
0286   31            050C  1415          BRW     FCNEXT                  ;SUCCESS RETURN
                     050F  1416 10$:
0268   31            050F  1417          BRW     FATALERO                ;
                     0512  1418
```

```
                        0512  1420 .SBTTL  POSITIONING FUNCTIONS
                        0512  1421
                        0512  1422 ;
                        0512  1423 ; SPACE FILE FORWARD
                        0512  1424 ; NOTE: HARDWARE SKIPFILE COMMAND IS NOT USED.
                        0512  1425 ; SKIPFILE IS SIMULATED BY A SERIES SKIPRECORD COMMANDS.
                        0512  1426 ;
                        0512  1427
                        0512  1428 SPCFILFOR:                                  ;SPACE FILE FORWARD
  7C A5   00B4 C5   B0  0512  1429         MOVW    UCB$W_MS_SPACNT(R5),UCB$W_BOFF(R5) ;SAVE NO. OF
  7E A5   00B4 C5   B0  0518  1430         MOVW    UCB$W_MS_SPACNT(R5),UCB$W_BCNT(R5) ; TAPE MARKS TO SKIP
00D0 C5   00B0 C5   D0  051E  1431         MOVL    UCB$L_RECORD(R5),UCB$L_MS_PMPR(R5) ;SAVE TAPE POSITION
  68 A5   0040 8F   A8  0525  1432         BISW    #UCB$M_MS_SWE,UCB$W_DEVSTS(R5)  ;USE OLD TAPE POSITION IF POWERFAIL
                        052B  1433                                         ;**SOFTWARE EMULATED FUNCTION**
                        052B  1434 0$:
      68 A5   01    AA  052B  1435         BICW    #UCB$M_MS_FEF,UCB$W_DEVSTS(R5)  ;CLEAR FLAG FOR 1ST EOF SEEN
                        052F  1436 1$:
0084 C5   7FFF 8F   B0  052F  1437         MOVW    #^X7FFF,UCB$W_MS_SPACNT(R5) ;SKIP 32,768 RECORDS INSTEAD
                        0536  1438         EXHC    10$,HC_STF                      ;DO IT
    51   00C4 C5   3C  053E  1439         MOVZWL  UCB$W_MS_XC(R5),R1      ;GET NO. OF RECORDS PASSED
    0080 C5   51   C0  0543  1440         ADDL    R1,UCB$L_RECORD(R5)     ;ADD IT
 DE 44 A5     11   E1  0548  1441         BBC     #MTS$V_EOF,UCB$L_DEVDEPEND(R5),0$ ;BR IF DIDN'T SEE TAPE MARK
         7E A5   B7   054D  1442         DECW    UCB$W_BCNT(R5)          ;DECREMENT TAPE MARK PASSED
                 13   E1  0550  1443         BBC     #DEV$V_MNT,-            ;BR IF NOT MOUNTED
      05 38 A5           0552  1444                 UCB$L_DEVCHAR(R5),2$    ;
                 18   E1  0555  1445         BBC     #DEV$V_FOR,-            ;BR IF MOUNTED NOT FOREIGN
      34 38 A5           0557  1446                 UCB$L_DEVCHAR(R5),5$    ;
                        055A  1447 2$:
   2B 68 A5   00    E1  055A  1448         BBC     #UCB$V_MS_FEF,UCB$W_DEVSTS(R5),4$ ;BR IF 1ST TMK
    00C4 C5   01    B1  055F  1449         CMPW    #1,UCB$W_MS_XC(R5)              ;**1 RECORD=TAPE MARK??**
              28    12  0564  1450         BNEQ    5$                              ;BR IF NO
    00B4 C5   01    B0  0566  1451         MOVW    #1,UCB$W_MS_SPACNT(R5)  ;SKIP 1 TMK REVERSE
                        056B  1452         EXHC    10$,HC_STR              ;
         7E A5    B6  0573  1453         INCW    UCB$W_BCNT(R5)          ;BACKUP 1 TMK PASSED
       00B0 C5    D7  0576  1454         DECL    UCB$L_RECORD(R5)        ;UPDATE TAPE POSITION
   50  09A0 8F    3C  057A  1455         MOVZWL  #SS$_ENDOFVOLUME,R0     ;YES, DOUBLE TMKS=ENDOFVOLUME
00C4 C5  7C A5  7E A5  A3  057F  1456         SUBW3   UCB$W_BCNT(R5),UCB$W_BOFF(R5),UCB$W_MS_XC(R5) ;GET NO. OF
                        0587  1457                                         ; TAPE MARKS PASSED
              020B  31  0587  1458         BRW     FCNEXT                  ;GO EXIT
                        058A  1459 4$:
      68 A5   01    A8  058A  1460         BISW    #UCB$M_MS_FEF,UCB$W_DEVSTS(R5) ;SET 1ST EOF
                        058E  1461 5$:
         7E A5    B5  058E  1462         TSTW    UCB$W_BCNT(R5)          ;PASSED ALL TAPE MARKS
              9C   12  0591  1463         BNEQ    1$                      ;NO, GO BACK
    00C4 C5   7C A5   B0  0593  1464         MOVW    UCB$W_BOFF(R5),UCB$W_MS_XC(R5) ;YES,COPY TMKS PASSED
              01F9  31  0599  1465         BRW     FCNEXT                  ;GO EXIT
                        059C  1466 10$:
              01DB  31  059C  1467         BRW     FATALERO                ;TAKE FAILURE RETURN
                        059F  1468
                        059F  1469 ;
                        059F  1470 ; SPACEFILE REVERSE
                        059F  1471 ;
                        059F  1472
                        059F  1473 SPCFILREV:                                  ;SPACE FILE REVERSE
  7C A5   00B4 C5   B0  059F  1474         MOVW    UCB$W_MS_SPACNT(R5),UCB$W_BOFF(R5) ;SAVE NO. OF
  7E A5   00B4 C5   B0  05A5  1475         MOVW    UCB$W_MS_SPACNT(R5),UCB$W_BCNT(R5) ; TAPE MARKS TO SKIP
00D0 C5   00B0 C5   D0  05AB  1476         MOVL    UCB$L_RECORD(R5),UCB$L_MS_PMPR(R5) ;SAVE TAPE POSITION
```

```
   68 A5   0040 8F   A8  05B2            BISW    #UCB$M_MS_SWE,UCB$W_DEVSTS(R5)  ;USE OLD TAPE POSITION IF POWERFAIL
                         05B8                                                   ;**SOFTWARE EMULATED FUNCTION**
                         05B8      1$:
 00B4 C5   7FFF 8F   B0  05B8            MOVW    #^X7FFF,UCB$W_MS_SPACNT(R5) ;SKIP 32,768 RECORDS INSTEAD
                         05BF            EXHC    10$,HC_STR                   ;DO IT
              10     E0  05C7            BBS     #MT$V_BOT,-                 ; If we ran into BOT, treat as if
        17 44 A5         05C9                    UCB$L_DEVDEPEND(R5),5$      ;  we were done.
   51   00C4 C5      3C  05CC            MOVZWL  UCB$W_MS_XC(R5),R1          ;GET NO. OF RECORDS PASSED
    00B0 C5   51     C2  05D1            SUBL    R1,UCB$L_RECORD(R5)         ;SUBTRACT
   DD 44 A5   11     E1  05D6            BBC     #MT$V_EOF,UCB$L_DEVDEPEND(R5),1$ ;BR IF DIDN'T SEE TAPE MARK
           7E A5     B7  05DB            DECW    UCB$W_BCNT(R5)              ;DECREMENT TAPE MARK PASSED
           7E A5     B5  05DE            TSTW    UCB$W_BCNT(R5)              ;PASSED ALL TAPE MARKS?
              D5     12  05E1            BNEQ    1$                          ;NO, BR BACK
                         05E3      5$:
           7E A5     A3  05E3            SUBW3   UCB$W_BCNT(R5),-            ; Calculate number of tape
           7C A5         05E6                    UCB$W_BOFF(R5),-            ;  marks passed.
         00C4 C5         05E8                    UCB$W_MS_XC(R5)
            01A7     31  05EB            BRW     FCNEXT                      ;GO EXIT
                         05EE      10$:
            0189     31  05EE            BRW     FATALERO
                         05F1
                         05F1      ;
                         05F1      ; SPACE RECORD FORWARD
                         05F1      ;
                         05F1
                         05F1      SPCRECFOR:                                ;SPACE RECORD FORWARD
                         05F1            EXHC    10$                         ;
   5C 44 A5   11     E1  05F6            BBC     #MT$V_EOF,UCB$L_DEVDEPEND(R5),8$ ;BR IF NO TMK
    00C4 C5   01     B1  05FB            CMPW    #1,UCB$W_MS_XC(R5)          ;**1 RECORD=TMK?**
              55     12  0600            BNEQ    8$                          ;BR IF NO
              13     E1  0602            BBC     #DEV$V_MNT,-                ;BR IF NOT MOUNTED
        05 38 A5         0604                    UCB$L_DEVCHAR(R5),2$        ;
              18     E1  0607            BBC     #DEV$V_FOR,-                ;BR IF MOUNTED NOT FOREIGN
        4B 38 A5         0609                    UCB$L_DEVCHAR(R5),8$        ;
                         060C      2$:
         00B0 C5     D5  060C            TSTL    UCB$L_RECORD(R5)            ;WAS AT BOT?
              45     13  0610            BEQL    8$                          ;BR IF YES
    00B4 C5   01     B0  0612            MOVW    #1,UCB$W_MS_SPACNT(R5)      ;SKIP 1 RECORD REVERSE
                         0617            EXHC    10$,HC_SRR                  ;
    00B4 C5   01     B0  061F            MOVW    #1,UCB$W_MS_SPACNT(R5)      ;SKIP 1 RECORD REVERSE
                         0624            EXHC    10$,HC_SRR                  ;
    00B4 C5   01     B0  062C            MOVW    #1,UCB$W_MS_SPACNT(R5)      ;SKIP 1 RECORD FORWARD
                         0631            EXHC    10$,HC_SRF                  ;
   0C 44 A5   11     E1  0639            BBC     #MT$V_EOF,UCB$L_DEVDEPEND(R5),6$ ;BR IF NO TMK
   50   09A0 8F      3C  063E            MOVZWL  #SS$_ENDOFVOLUME,R0         ;WAS AT ENDOFVOLUME
         00C4 C5     B4  0643            CLRW    UCB$W_MS_XC(R5)             ;NO RESULTANT MOVEMENT
            014B     31  0647            BRW     FCNEXT                      ;RETURN
                         064A      6$:
    00B4 C5   01     B0  064A            MOVW    #1,UCB$W_MS_SPACNT(R5)      ;SKIP 1 RECORD FORWARD
                         064F            EXHC    10$,HC_SRF                  ;
                         0657      8$:
   51   00C4 C5      3C  0657            MOVZWL  UCB$W_MS_XC(R5),R1          ;GET NO. OF RECORDS PASSED
    00B0 C5   51     C0  065C            ADDL    R1,UCB$L_RECORD(R5)         ;UPDATE
            0131     31  0661            BRW     FCNEXT                      ;
                         0664      10$:
            0113     31  0664            BRW     FATALERO
                         0667
```

```
                     0667  1534 ;
                     0667  1535 ; SPACE RECORD REVERSE
                     0667  1536 ;
                     0667  1537
                     0667  1538 SPCRECREV:                                  ;SPACE RECORD REVERSE
                     0667  1539          EXHC    10$                       ;
          10   E0    066C  1540          BBS     #MT$V_BOT,-               ; If we ran into BOT, treat as if
     0A 44 A5         066E  1541                  UCB$L_DEVDEPEND(R5),5$    ;  we were done.
51   00C4 C5    3C   0671  1542          MOVZWL  UCB$W_MS_XC(R5),R1        ;GET NO. OF RECORDS PASSED
0080 C5    51    C2   0676  1543          SUBL    R1,UCB$L_RECORD(R5)       ;UPDATE
                     067B  1544 5$:
          0117   31   067B  1545          BRW     FCNEXT                    ;
                     067E  1546 10$:
          00F9   31   067E  1547          BRW     FATALERO
                     0681  1548
                     0681  1549 ;
                     0681  1550 ; REWIND
                     0681  1551 ;
                     0681  1552
                     0681  1553 REWIND:                                     ;REWIND
                     0681  1554          EXHC    10$
  46 A5    01   A8   0686  1555          BISW    #<MT$M_BOT@-16>,UCB$L_DEVDEPEND+2(R5) ;MARK BOT
  46 A5    10   AA   068A  1556          BICW    #<MT$M_LOST@-16>,UCB$L_DEVDEPEND+2(R5) ;CLEAR POSITION-LOST
     0080 C5   D4   068E  1557          CLRL    UCB$L_RECORD(R5)
          0100   31   0692  1558          BRW     FCNEXT
                     0695  1559 10$:
          00E2   31   0695  1560          BRW     FATALERO
                     0698  1561
```

```
                     0698  1563 .SBTTL  FORMAT COMMANDS
                     0698  1564
                     0698  1565 ;
                     0698  1566 ; WRITE TAPE MARK
                     0698  1567 ;
                     0698  1568
                     0698  1569 WRTTMK:                                          ;WRITE TAPE MARK
 46 A5   08   AA     0698  1570          BICW    #<MT$M_HWL>@-16,UCB$L_DEVDEPEND+2(R5) ;CLEAR
                     069C  1571                                                  ; WRITE LOCK BIT FIRST
                     069C  1572          EXHC    10$                             ;
   00B0 C5   D6      06A1  1573          INCL    UCB$L_RECORD(R5)                ;INCREMENT RECORD COUNT
      00ED   31      06A5  1574          BRW     FCNEXT                          ;GOTO EXIT
                     06A8  1575 10$:
        50   DD      06A8  1576          PUSHL   R0                              ;SAVE R0
00000000'GF   16     06AA  1577          JSB     G^ERL$DEVICERR                  ;LOG BEFORE RETRY
        50 8ED0      06B0  1578          POPL    R0                              ;RESTORE
                     06B3  1579 20$:
   50   04   D1      06B3  1580          CMPL    #TCC_REM,R0                     ;TAPE MOVED?
        15   13      06B6  1581          BEQL    22$                             ;YES
   0080 C5   97      06B8  1582          DECB    UCB$B_ERTCNT(R5)                ;ANY RETRIES LEFT?
        24   19      06BC  1583          BLSS    30$                             ;NO, FATAL
                     06BE  1584          EXHC    20$,MC_WTM                      ;DO IT AGAIN
   00B0 C5   D6      06C6  1585          INCL    UCB$L_RECORD(R5)                ;INCREMENT RECORD COUNT
      00C8   31      06CA  1586          BRW     RFCNEXT                         ;RETURN
                     06CD  1587 22$:
   0080 C5   97      06CD  1588          DECB    UCB$B_ERTCNT(R5)                ;ANY RETRIES LEFT?
        0F   19      06D1  1589          BLSS    30$                             ;NO, FATAL
                     06D3  1590          EXHC    22$,MC_WTR                      ;DO WRITE TAPE MARK RETRY
   00B0 C5   D6      06DB  1591          INCL    UCB$L_RECORD(R5)                ;INCREMENT RECORD COUNT
      00B3   31      06DF  1592          BRW     RFCNEXT                         ;
                     06E2  1593
      0095   31      06E2  1594 30$:     BRW     FATALERO                        ;BRANCH FATAL ERROR
                     06E5  1595 ;
                     06E5  1596 ; WRITE TAPE MARK RETRY(SPACE REV,ERASE,WRITE TAPE MARK)
                     06E5  1597 ;
                     06E5  1598
                     06E5  1599 WRTTMKR:                                                 ;WRITE TAPE MARK RETRY
                     06E5  1600          EXHC    10$                             ;
   00A8   31         06EA  1601          BRW     FCNEXT                          ;GO EXIT
                     06ED  1602 10$:
   008A   31         06ED  1603          BRW     FATALERO                        ;FATAL AS NOW
                     06F0  1604 ;
                     06F0  1605 ; ERASE
                     06F0  1606 ;
                     06F0  1607
                     06F0  1608 ERASE:                                           ;ERASE
                     06F0  1609          EXHC    10$                                     ;
   009D   31         06F5  1610          BRW     FCNEXT
                     06F8  1611 10$:
   007F   31         06F8  1612          BRW     FATALERO
                     06FB  1613
```

```
                     06FB  1615 .SBTTL  CONTROL COMMANDS
                     06FB  1616
                     06FB  1617 ;
                     06FB  1618 ; CONTROL COMMANDS
                     06FB  1619 ;
                     06FB  1620
                     06FB  1621 MSGREL:                                     ;MESSAGE BUFFER RELEASE
                     06FB  1622         EXHC    10$                                 ;
0092   31            0700  1623         BRW     FCNEXT
                     0703  1624 10$:
                     0703  1625
0074   31            0703  1626         BRW     FATALERO
                     0706  1627
                     0706  1628 UNLOAD:                                     ;
                     0706  1629         EXHC    10$                                 ;
0087   31            0708  1630         BRW     FCNEXT
                     070E  1631 10$:
0069   31            070E  1632         BRW     FATALERO
                     0711  1633
                     0711  1634 CLEAN:                                      ;CLEAN
                     0711  1635         EXHC    10$                                 ;
007C   31            0716  1636         BRW     FCNEXT
                     0719  1637 10$:
005E   31            0719  1638         BRW     FATALERO
                     071C  1639
```

```
                                                    K 5
                     071C  1641 .SBTTL .INITIALIZE AND GET STATUS
                     071C  1642
                     071C  1643 ;
                     071C  1644 ; DRIVE INITIALIZE
                     071C  1645 ;
                     071C  1646
                     071C  1647 DRVCLR:                                         ;DRIVE INITIALIZE
                     071C  1648         EXHC    10$
                0071 31   0721  1649         BRW     FCNEXT
                     0724  1650 10$:
                0053 31   0724  1651         BRW     FATALERO
                     0727  1652
                     0727  1653 ;
                     0727  1654 ; GET STATUS (END MESSAGE ONLY)
                     0727  1655 ;
                     0727  1656
                     0727  1657 GETSTS:                                         ;
                     0727  1658         EXHC    10$
            46 A5 0F AA 072C  1659         BICW    #<MT$M_BOT!-                ;CLEAR BITS IN UCB$L_DEVDEPEND+2
                     0730  1660                 MT$M_EOF!-                  ;
                     0730  1661                 MT$M_EOT!-                  ;END OF TAPE
                     0730  1662                 MT$M_HWL>@-16,UCB$L_DEVDEPEND+2(R5) ;
                  01 E1 0730  1663         BBC     #MS_XSR0_V_BOT,-            ;AT BOT?
            0C 00FE C5      0732  1664                 UCB$W_MS_XSR0(R5),1$                ;BR IF NO
               00B0 C5 D4   0736  1665         CLRL    UCB$L_RECORD(R5)            ;CLEAR RECORD COUNT
            46 A5 01 A8 073A  1666         BISW    #<MT$M_BOT@-16>,UCB$L_DEVDEPEND+2(R5) ;SET BOT
            46 A5 10 AA 073E  1667         BICW    #<MT$M_LOST@-16>,UCB$L_DEVDEPEND+2(R5) ;CLEAR LOST BIT
                     0742  1668 1$:
                  0F E1 0742  1669         BBC     #MS_XSR0_V_TMK,-            ;AT TAPE MARK
            04 00FE C5      0744  1670                 UCB$W_MS_XSR0(R5),2$                ;BR IF NO
            46 A5 02 A8 0748  1671         BISW    #<MT$M_EOF@-16>,UCB$L_DEVDEPEND+2(R5) ;SET EOF
                     074C  1672 2$:
                  02 E1 074C  1673         BBC     #MS_XSR0_V_WLK,-            ;WRITE-LOCKED?
            04 00FE C5      074E  1674                 UCB$W_MS_XSR0(R5),3$                ;BR IF NO
            46 A5 08 A8 0752  1675         BISW    #<MT$M_HWL@-16>,UCB$L_DEVDEPEND+2(R5) ;SET WRITE-LOCKED
                     0756  1676 3$:
                  00 E1 0756  1677         BBC     #MS_XSR0_V_EOT,-            ;END OF TAPE
            0D 00FE C5      0758  1678                 UCB$W_MS_XSR0(R5),4$                ;BR IF NO
            46 A5 10 AA 075C  1679         BICW    #<MT$M_LOST@-16>,UCB$L_DEVDEPEND+2(R5) ;CLEAR POS.LOST
            46 A5 04 A8 0760  1680         BISW    #<MT$M_EOT@-16>,UCB$L_DEVDEPEND+2(R5) ;SET END OF TAPE
         50 0878 8F 3C 0764  1681         MOVZWL  #SS$_ENDOFTAPE,R0           ;PUT IN RETURN STATUS
                     0769  1682 4$:
                  06 E0 0769  1683         BBS     #MS_XSR0_V_ONL,-            ;CHECK IF ONLINE?
            05 00FE C5      076B  1684                 UCB$W_MS_XSR0(R5),6$        ;BR IF YES
                     076F  1685
         50 01A4 8F 3C 076F  1686         MOVZWL  #SS$_MEDOFL,R0              ;RETURN MEDIUM-OFFLINE
                     0774  1687 6$:
                001E 31   0774  1688         BRW     FCNEXT
                     0777  1689 10$:
                0000 31   0777  1690         BRW     FATALERO                    ;TREAT AS FATAL
                     077A  1691
```

```
                     077A  1693 .SBTTL  COMPLETION PROCESSING
                     077A  1694 ;
                     077A  1695 ; FATALERR - FINISHING UP THE I/O REQUEST PROCESSING WHEN THE OPERATION
                     077A  1696 ; ENDS WITH FATAL OR HARD ERROR.
                     077A  1697 ; R0 HAS THE FINAL STATUS CODE ALREADY.
                     077A  1698 ;
                     077A  1699
                     077A  1700 FATALERO:                                    ;NO ERROR CODE IN R0
          50  008C BF   3C  077A  1701          MOVZWL  #SS$_DRVERR,R0       ;GIVE IT ONE FOR NOW
                     077F  1702 FATALERR:
          00C4 C5    B4  077F  1703          CLRW    UCB$W_MS_XC(R5)      ;MAKE SURE NOTHING XFERRED/SKIPPED
                     0783  1704
          50  01A4 BF   B1  0783  1705          CMPW    #SS$_MEDOFL,R0       ; See if error is MEDIA OFF LINE.
                08   13  0788  1706          BEQL    FCNEXT               ; If so, then branch around logging error.
                50   DD  078A  1707          PUSHL   R0                   ;SAVE FINAL STATUS
     00000000'GF   16  078C  1708          JSB     G^ERL$DEVICERR       ;LOG DEVICE ERROR
                50 8ED0  0792  1709          POPL    R0                   ;
                     0795  1710 RFCNEXT:                                   ;SUCCESS RETURN AFTER RETRY
                     0795  1711 FCNEXT:
68 A5       0840 8F   AA  0795  1712          BICW    #<UCB$M_MS_RPI!UCB$M_MS_SWE>,UCB$W_DEVSTS(R5) ;ASSURE FLAGS CLEARED
                50   DD  079B  1713          PUSHL   R0                   ;SAVE FINAL STATUS
     00000000'GF   16  079D  1714          JSB     G^IOC$DIAGBUFILL     ;FILL DIAGNOSTIC BUFFER IF PRESENT
02 AE    00C4 C5   B0  07A3  1715          MOVW    UCB$W_MS_XC(R5),2(SP) ;SET BYTES XFERRED OR RECORDS/FILES SKIPED
           36 6E   E8  07A9  1716          BLBS    (SP),70$             ;IF LBS SUCCESSFUL COMPLETION
        54   58 A5   D0  07AC  1717          MOVL    UCB$L_IRP(R5),R4     ;GET ADDRESS OF CURRENT I/O PACKET
  2D 2A A4      04   E1  07B0  1718          BBC     #IRP$V_VIRTUAL,IRP$W_STS(R4),70$ ;IF CLR, NOT VIRTUAL FUNCTION
        54   18 A4   D0  07B5  1719          MOVL    IRP$L_WIND(R4),R4    ;GET ADDRESS OF WINDOW BLOCK
             16 A4   B4  07B9  1720          CLRW    WCB$W_NMAP(R4)       ;CLEAR NUMBER OF MAPPING POINTERS
        54   34 A5   D0  07BC  1721          MOVL    UCB$L_VCB(R5),R4     ;GET ADDRESS OF VCB LISTHEAD
        52   4C A5   9E  07C0  1722          MOVAB   UCB$L_IOQFL(R5),R2   ;GET ADDRESS OF I/O QUEUE
             53   52   D0  07C4  1723          MOVL    R2,R3                ;SET ADDRESS OF PREVIOUS ENTRY
             53   63   D0  07C7  1724 60$:     MOVL    (R3),R3              ;GET ADDRESS OF NEXT ENTRY
             52   53   D1  07CA  1725          CMPL    R3,R2                ;END OF LIST?
                13   13  07CD  1726          BEQL    70$                  ;IF EQL YES
  F3 2A A3      04   E1  07CF  1727          BBC     #IRP$V_VIRTUAL,IRP$W_STS(R3),60$ ;IF CLR,NOT VIRTUAL FUNCTION
        53   04 A3   D0  07D4  1728          MOVL    4(R3),R3                     ;RETRIEVE ADDRESS OF PREVIOUS ENTRY
        51   00 B3   0F  07D8  1729          REMQUE  @(R3),R1             ;REMOVE ENTRY FROM DRIVER QUEUE
     04 B4      61   0E  07DC  1730          INSQUE  (R1),@4(R4)          ;INSERT ENTRY IN BLOCKED I/O LIST
                E5   11  07E0  1731          BRB     60$                  ;
                50 8ED0  07E2  1732 70$:     POPL    R0                   ;RETRIEVE FINAL STATUS
        51   44 A5   D0  07E5  1733          MOVL    UCB$L_DEVDEPEND(R5),R1 ;SET MAGTAPE STATUS AND CHARACTERISTIC
                     07E9  1734          .IF     DF      TS_TRACE
                     07E9  1735          BSBW    TRACE_STATUS             ; Trace final I/O status.
                     07E9  1736          .ENDC
                     07E9  1737          REQCOM                           ;COMPLETE REQUEST
                     07EF  1738
```

```
                     07EF  1740 .SBTTL  HARDWARE COMMAND EXECUTOR
                     07EF  1741 ;+
                     07EF  1742 ; HCEX - EXECUTES HARDWARE COMMAND
                     07EF  1743 ;
                     07EF  1744 ; THIS ROUTINE IS CALLED VIA A BSB WITH A WORD IMMEDIATELY FOLLOWING THAT
                     07EF  1745 ; SPECIFIES THE ADDRESS OF AN (RETRIABLE) ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE
                     07EF  1746 ; BEEN SET UP IN THE UCB BEFORE THE CALL. THE COMMAND PACKET IS APPROPRIEATELY
                     07EF  1747 ; SETUP AND INITIATED BY LOADING THE ADDR. OF THE COMMAND PACKET INTO THE
                     07EF  1748 ; TS11/TS04 DEVICE REGISTER, TSDB. THEN, A WAITFOR INTERRUPT IS EXECUTED
                     07EF  1749 ; AND WHEN THE INTERRUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.
                     07EF  1750 ; THE ROUTINE MAINLY DEALS WITH THE HARDWARE INTERFACE.
                     07EF  1751 ;
                     07EF  1752 ; INPUTS:
                     07EF  1753 ;       R0=HARDWARE COMMAND TABLE DISPATCH INDEX
                     07EF  1754 ;       R4=EQUIVALENT CSR ADDR. FOR TS11/TS04
                     07EF  1755 ;       R5=DEVICE UNIT UCB ADDRESS
                     07EF  1756 ;
                     07EF  1757 ;       00(SP) = RETURN ADDRESS OF CALLER
                     07EF  1758 ;       04(SP) = RETURN ADDRESS OF CALLER'S CALLER
                     07EF  1759 ;
                     07EF  1760 ;       IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A WORD WHICH HAS
                     07EF  1761 ;       A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE, IF APPROPRIATE.
                     07EF  1762 ; OUTPUTS:
                     07EF  1763 ;       THE DRIVE STATUS, SUCH AS BOT, EOT, ETC, ARE RECORDED IN UCB.
                     07EF  1764 ;
                     07EF  1765 ;       THERE ARE THREE EXITS FROM THIS ROUTINE:
                     07EF  1766 ;       1) NORMAL RETURN,
                     07EF  1767 ;       2) FATAL OR HARD ERROR  EXIT, AND
                     07EF  1768 ;       3) RETRIABLE ERROR RETURN.
                     07EF  1769 ; WHEN EXITS, R0 HAS THE FINAL STATUS CODE IF NORMAL OR FATAL,
                     07EF  1770 ; R0 HAS TERMINATION CODE, 4 OR 5, IF RETRIABLE.
                     07EF  1771 ; THE DRIVE STATUS IS RECORDED INTO UCB WHILE PROCESSING TERMINATION CODE
                     07EF  1772 ;
                     07EF  1773
                     07EF  1774 HCEX:                                                ;
     009C C5 8ED0    07EF  1775          POPL    UCB$L_DPC(R5)                       ;SAVE DRIVER PC VALUE
00DC C5    14   9A   07F4  1776          MOVZBL  #20,UCB$L_MS_TIMOUT(R5)             ; Initialize timeout to 20 seconds.
     00C6 C5    B4   07F9  1777          CLRW    UCB$B_MS_DPN(R5)                    ;CLEAR DATA PATH NO. & PURGE ERROR
     00C8 C5    7C   07FD  1778          CLRQ    UCB$L_MS_DPR(R5)                    ;ZERO DATAPATH REG. & FINAL MAP REG.
     00C4 C5    B4   0801  1779          CLRW    UCB$W_MS_XC(R5)                     ;INITIALIZE COUNT
0093 C5    50   90   0805  1780          MOVB    R0,UCB$B_CEX(R5)                    ;SAVE CASE INDEX
  51   00B6 C5    D0   080A  1781          MOVL    UCB$L_MS_TSPT1(R5),R1               ;GET COMMAND PACKET POINTER
61   F824 CF40    B0   080F  1782          MOVW    HCTAB[R0],MS_CPHD(R1)               ;LOAD COMMAND PACKET HEAD WORD
```

```
                10 A1   FFFF BF    B0   0815  1784          MOVW    #^XFFFF,MS_MHD(R1)      ;MARK MSG HEAD TO ENSURE MSG BUFFER RETURNED
                  05 68 A5   0C    E5   081B  1785          BBCC    #UCB$V_MS_VCK,UCB$W_DEVSTS(R5),7$ ;BR IF NOT VOLUME CHECK
                    61   4000 BF    A8   0820  1786          BISW    #MS_CPRD_M_CVC,MS_CPHD(R1) ;YES, FLAG TO CLEAR VOLUME CHECK
                                          0825  1787 7$:
                                          0825  1788          CASE    R0,<-                   ;DISPATCH TO PROPER COMMAND ROUTINE
                                          0825  1789                  PNOP,-                  ;NOP
                                          0825  1790                  PMIS,-                  ;UNLOAD
                                          0825  1791                  PPOS,-                  ;SPACE FILE FORWARD
                                          0825  1792                  PPOS,-                  ;REWIND
                                          0825  1793                  PMIS,-                  ;DRIVE CLEAR
                                          0825  1794                  PPOS,-                  ;SPACE FILE REVERSE
                                          0825  1795                  PMIS,-                  ;ERASE
                                          0825  1796                  PPOS,-                  ;SPACE RECORD REVERSE
                                          0825  1797                  PNOP,-                  ;SIMULATED PACK ACKNOWLEDGE
                                          0825  1798                  PPOS,-                  ;SPACE RECORD FORWARD
                                          0825  1799                  PNOP,-                  ;SIMULATED WRITECHECK
                                          0825  1800                  PXFR,-                  ;WRITE DATA FORWARD
                                          0825  1801                  PXFR,-                  ;READ DATA FORWARD
                                          0825  1802                  PNOP,-                  ;SIMULATED WRITE CHECK REVERSE
                                          0825  1803                  PXFR,-                  ;WRITE DATA (NO REVERSE)
                                          0825  1804                  PXFRR,-                 ;READ DATA REVERSE
                                          0825  1805                  PXFRRD,-                ;REREAD DATA NEXT
                                          0825  1806                  PXFRRD,-                ;REREAD DATA PREVIOUS
                                          0825  1807                  PXFR,-                  ;WRITE DATA RETRY
                                          0825  1808                  PNOP,-                  ;SIMULATED READ PRESET
                                          0825  1809                  PNOP,-                  ;SIMULATED SET CHARACTERISTIC
                                          0825  1810                  PMIS,-                  ;GET STATUS IMMEDIATE(SENS CHAR.)
                                          0825  1811                  PMIS,-                  ;WRITE TAPE MARK
                                          0825  1812                  PMIS,-                  ;WRITE TAPE MARK RETRY
                                          0825  1813                  PMIS,-                  ;CLEAN
                                          0825  1814                  PMIS,-                  ;MESSAGE BUFFER RELEASE
                                          0825  1815                  PXFR,-                  ;WRITE SUBSYSTEM MEMORY
                                          0825  1816                  PWCH,-                  ;WRITE CHARACTERISTICS
                                          0825  1817                  >
                                          0861  1818
```

```
                        0861  1820 ;+PNOP - NO OPERATION ON HARDWARE FOR MANY SIMULATED FUNCTIONS
                        0861  1821 ;       THEY ARE: NOP,PACK ACKNOWLEDGE,WRITECHECK,WRITE CHECK REVERSE,
                        0861  1822 ;       READ IN PRESET,SET CHARACTERISTICS.
                        0861  1823 ;       THE ROUTINE SIMPLY RETURNS.
                        0861  1824
                        0861  1825 PNOP:                                  ;
                        0861  1826 RET:                                   ;
          50   01   3C   0861  1827         MOVZWL  #SS$_NORMAL,R0        ;ALWAYS SUCCESS
009C C5     02   C0   0864  1828         ADDL    #2,UCB$L_DPC(R5)      ;ADJUST TO CORRECT RETURN
      009C D5   17   0869  1829         JMP     @UCB$L_DPC(R5)        ;RETURN TO DRIVER
                        086D  1830
                        086D  1831
                        086D  1832 ; PPOS - CONSTRUCT COMMNAD PACKET FOR POSITIONING COMMANDS:
                        086D  1833 ; SPACE RECORDS FORWARD, SPACE RECORDS REVERSE, SKIP TAPE MARKS FORWARD,
                        086D  1834 ; SPACE TAPE MARKS REVERSE, AND REWIND.
                        086D  1835
                        086D  1836 PPOS:                                  ;
          50   03   91   086D  1837         CMPB    #CDHC_RWD,R0          ;REWIND COMMAND?
                2A   12   0870  1838         BNEQ    5$                    ;NO
                        0872  1839         DSBINT                        ;YES, DISABLE INTERRUPTS
                05   E1   0878  1840         BBC     #UCB$V_POWER,-
                                                UCB$W_STS(R5),1$          ; Continue if NO powerfail
          03 64 A5          087A  1841
                0090   31   087D  1842         BRW     PWRFLT                ; Branch around if we had POWERFAIL.
                        0880  1843 1$:
       64   00BE C5   B0   0880  1844         MOVW    UCB$W_MS_TSPT3(R5),(R4) ;LOAD COMMAND POINTER
                        0885  1845         WFIKPCH MSTMOT,#^D300         ;TIMEOUT FOR 5 MIN.'S FOR 2400 FEET TAPE
                        0893  1846         IOFORK                        ;MAKE IT FORK FIRST
                0193   31   0899  1847         BRW     XTC                   ;GO PROCESS TERMINATION CODE
                        089C  1848 5$:                                    ;NOT REWIND OR UNLOAD
    02 A1   00B4 C5   B0   089C  1849         MOVW    UCB$W_MS_SPACNT(R5),MS_BACT(R1) ;LOAD THE COUNT
00C4 C5     00B4 C5   B0   08A2  1850         MOVW    UCB$W_MS_SPACNT(R5),UCB$W_MS_XC(R5) ;COPY COUNT
                        08A9  1851                                        ;FALL INTO CODE TO LOAD COMMAND
          00B4 C5   3C   08A9  1852         MOVZWL  UCB$W_MS_SPACNT(R5),-     ; Maximum size record takes one second
                00DC C5         08AD  1853                                    UCB$L_MS_TIMOUT(R5)       ;  to skip (approximately).
      00000294 8F   D1   08B0  1854         CMPL    #11*60,-                  ; Compare time to skip entire tape to
                00DC C5         08B6  1855                                    UCB$L_MS_TIMOUT(R5)       ;  time to skip this # records.
                09   18   08B9  1856         BGEQ    10$                       ; GEQ implies skipping small # records
          0294 8F   3C   08BB  1857         MOVZWL  #11*60,-                  ; Else use maximum time to skip whole
                00DC C5         08BF  1858                                    UCB$L_MS_TIMOUT(R5)       ;  tape.
                05   11   08C2  1859         BRB     20$                       ; And branch around.
     00DC C5     02   C0   08C4  1860 10$:    ADDL    #2,UCB$L_MS_TIMOUT(R5)    ; Add in fudge factor for small skips.
                        08C9  1861 20$:
                        08C9  1862
                        08C9  1863 ;+PMIS - CONSTRUCT COMMAND PACKET FOR FORMAT COMMANDS: WRITE TAPE MARK,
                        08C9  1864 ; ERASE,WRITE TAPE MARK RETRY; CONTROL COMMANDS: MESSAGE BUFFER RELEASE,
                        08C9  1865 ; UNLOAD, & CLEAN; INITIALIZE COMMAND: DRIVE INITIALIZE; AND GET STATUS
                        08C9  1866 ; COMMAND: GET STATUS IMMEDIATE.
                        08C9  1867
                        08C9  1868 PMIS:                                  ;
                0U1F   31   08C9  1869         BRW     LDTSDB                ;GO LOAD DEVICE REGISTER
                        08CC  1870
                        08CC  1871 ; PWCH - CONSTRUCT COMMAND PACKET FOR WRITE CHARACTERISTIC COMMAND
                        08CC  1872
                        08CC  1873 PWCH:                                  ; R1 POINTS TO COMMAND PACKET
  02 A1   00BA C5   D0   08CC  1874         MOVL    UCB$L_MS_TSPT2(R5),MS_BACT(R1) ;STORE CHAR. BUFFER ADDR.
      02 A1   08   C0   08D2  1875         ADDL    #8,MS_BACT(R1)            ;POINT TO CHAR. BUFFER NOW
      06 A1   08   B0   08D6  1876         MOVW    #8,MS_CNT(R1)             ;STORE BYTE COUNT FOR CHAR. DATA
```

```
                08 A1   00BA C5   D0   08DA  1877          MOVL    UCB$L_MS_TSPT2(R5),MS_MBA0(R1)          ;STORE MESSAGE BUFFER ADDR.
                   08 A1    10   C0   08E0  1878          ADDL    #16,MS_MBA0(R1)                         ; AS CHAR. DATA
                   0C A1    0E   B0   08E4  1879          MOVW    #14,MS_LNTH(R1)                         ;LENGTH OF CHAR. DATA=14.
                      0E A1     B4   08E8  1880          CLRW    MS_CHWD(R1)                             ;ZERO CHARACTERISTIC WORD
                                     08EB  1881 ;
                                     08EB  1882 ; NOW, COMMAND PACKET IS SETUP, READY TO LOAD DEVICE REGISTER
                                     08EB  1883 ;
                                     08EB  1884
                                     08EB  1885 LDTSDB:                                                ;TS11/TS04 CSR EQUIVALENT=TSDB
                                     08EB  1886          DSBINT                                        ;DISABLE INTERRUPTS
                1A 64 A5    05   E0   08F1  1887          BBS     #UCB$V_POWER,UCB$W_STS(R5),PWRFL1 ;BR IF POWERFAILED
                 64    0BE C5    B0   08F6  1888          MOVW    UCB$W_MS_TSPT3(R5),(R4) ;LOAD THE COMMAND POINTER
                                     08FB  1889          WFIKPCH MSTMOT,UCB$L_MS_TIMOUT(R5)
                                     0907  1890          IOFORK                                        ;MAKE IT FORK FIRST
                      011F    31   090D  1891          BRW     XTC                                   ;PROCESS TERMINATION CODE
                                     0910  1892
                                     0910  1893 ;
                                     0910  1894 ; HERE, TREAT POWERFAIL AS TIMEOUT
                                     0910  1895 ;
                                     0910  1896
                                     0910  1897 PWRFL1:
                                     0910  1898          ENBINT                                        ;ENABLE INTERRUPTS
                      02DD    31   0913  1899          BRW     MSTMO1                                ;GOTO TIMEOUT ROUTINE
                                     0916  1900
                                     0916  1901 ; PXFR - CONSTRUCT COMMAND PACKET FOR DATA TRANSFER COMMANDS:
                                     0916  1902 ; READ NEXT(FORWARD), READ PREVIOUS(REVERSE), REREAD PREVIOUS(SPACE REV, READ
                                     0916  1903 ; FWD), REREAD NEXT(SPACE FWD, READ REV), WRITE DATA, WRITE DATA RETRY,
                                     0916  1904 ; AND WRITE SUBSYSTEM MEMORY.
                                     0916  1905
                                     0916  1906 PXFRRD:                                                ; REREAD COMMANDS ENTER HERE.
             05 009A C5     09   E1   0916  1907          BBC     #IO$V_OPPOSITE, -                     ; Branch if opposite bit not set.
                                     091C  1908                  UCB$W_FUNC(R5), 10$
                61    2000 8F     A8   091C  1909          BISW    #MS_CPHD_M_OPP, -                     ; If its set, propogate it to the
                                     0921  1910                  MS_CPHD(R1)                           ; command header.
                         0A     11   0921  1911 10$:      BRB     PXFRR                                 ; Then rejoin common code.
                                     0923  1912
                                     0923  1913 PXFR:                                                  ;
                   68 A5     20   A8   0923  1914          BISW    #UCB$M_MS_RDPR,UCB$W_DEVSTS(R5) ;FLAG BUFFERED DATAPATH
                                     0927  1915          REQDPR                                        ;REQUEST DATAPATH
                                     092D  1916 PXFRR:                                ;*ENTRY PT FOR READ REVERSE*
                                     092D  1917                                       ;**WHICH USES DIRECT DATA PATH**
                                     092D  1918          REQMPR                                        ;REQUEST MAP REGISTER
                                     0933  1919          LOADUBAA                                      ;LOAD MAP REGISTER
                   50    7C A5     3C   0939  1920          MOVZWL  UCB$W_BOFF(R5),R0                 ;GET BYTE OFFSET
                   51    24 A5     D0   093D  1921          MOVL    UCB$L_CRB(R5),R1                  ;GET CRB
          50    09    09    34 A1    F0   0941  1922          INSV    CRB$L_INTD+VEC$W_MAPREG(R1),#9,#9,R0 ;INSERT HGH 9 BITS
                51    00B6 C5     D0   0947  1923          MOVL    UCB$L_MS_TSPT1(R5),R1             ;GET COMMAND PACKET ADDR.
                   02 A1     50   D0   094C  1924          MOVL    R0,MS_BACT(R1)                    ;STORE XFR ADDR.
                06 68 A5     01   E0   0950  1925          BBS     #UCB$V_MS_SWAP,UCB$W_DEVSTS(R5),12$ ;BR IF INDUSTRI.COMP.
                                     0955  1926                                                    ; (SET BY SETCHAR COMMAND)
             05 009A C5     08   E1   0955  1927          BBC     #IO$V_SWAP,UCB$W_FUNC(R5),15$ ;SWAP BIT SET??
                                     095B  1928 12$:
                61    1000 8F     A8   095B  1929          BISW    #MS_CPHD_M_SWB,MS_CPHD(R1) ;YES, SET IT IN CMD HEADER
                                     0960  1930 15$:
                06 A1     7E A5     B0   0960  1931          MOVW    UCB$W_BCNT(R5),MS_CNT(R1) ;STORE BYTE COUNT
             00C4 C5      7E A5     B0   0965  1932          MOVW    UCB$W_BCNT(R5),UCB$W_MS_XC(R5) ;COPY BYTE COUNT
                                     096B  1933          DSBINT                                        ;DISABLE INTERRUPTS
```

```
                4B 64 A5    05   E0  0971  1934          BBS     #UCB$V_POWER,UCB$W_STS(R5),20$ ;BR IF POWERFAILED
                 64   00BE C5   B0  0976  1935          MOVW    UCB$W_MS_TSPT3(R5),(R4) ;LOAD THE COMMAND POINTER
                                  097B  1936          WFIKPCH MSTMO,#^D20             ;TIMEOUT FOR 20. SEC.'S
                                  0985  1937          IOFORK                  ;
                  00000000'GF   16   098B  1938          JSB     G^IOC$PURGDATAP         ;PURGE DATAPATH
                        05 50   E8   0991  1939          BLBS    R0,18$                  ;BR IF NO ERROR
                 00C7 C5    01   90   0994  1940          MOVB    #1,UCB$B_MS_PER(R5)     ;MARK PURGE ERROR
                                  0999  1941  18$:
                26 64 A5    05   E0  0999  1942          BBS     #UCB$V_POWER,UCB$W_STS(R5),21$ ;BR IF POWERFAILED
                                  099E  1943          ;** WANT TO SAVE DATAPATH REG. & NO. & ETC. FOR ERRLOG
            00C4 C5    00FC C5   A2  099E  1944          SUBW    UCB$W_MS_RBPC(R5),UCB$W_MS_XC(R5) ;GET BYTES ACTUALLY XFERRED
                 00C8 C5    51   D0  09A5  1945          MOVL    R1,UCB$L_MS_DPR(R5)     ;GET DATA PATH REGISTER
                       37 A3    90  09AA  1946          MOVB    CRB$L_INTD+VEC$B_DATAPATH(R3),-
                    00C6 C5         09AD  1947                  UCB$B_MS_DPN(R5)        ;GET DATA PATH NO.
                      51   7C A5   3C  09B0  1948          MOVZWL  UCB$W_BOFF(R5),R1       ;GET BYTE OFFSET
                    00C6 C5    95   09B4  1949          TSTB    UCB$B_MS_DPN(R5)        ;DATAPATH NO.=0(DIRECT)?
                         0D    12   09B8  1950          BNEQ    22$                     ;NO
                      51 00FC C5   A0  09BA  1951          ADDW    UCB$W_MS_RBPC(R5),R1            ;YES, REVERSE OPERATION
                         0B    11   09BF  1952          BRB     24$                     ;SKIP AROUND
                                  09C1  1953
                                  09C1  1954  ;
                                  09C1  1955  ; HERE,TREAT POWERFAIL AS TIMEOUT
                                  09C1  1956  ;
                                  09C1  1957
                                  09C1  1958  20$:
                                  09C1  1959          ENBINT                          ;ENABLE INTERRUPTS
                                  09C4  1960  21$:
                       0215   31   09C4  1961          BRW     MSTMO
                                  09C7  1962
                                  09C7  1963  22$:
                   51   00C4 C5   A0  09C7  1964          ADDW    UCB$W_MS_XC(R5),R1      ;GET FINAL ADDR. OFFSET
                                  09CC  1965  24$:
                51   51    F7 8F   78  09CC  1966          ASHL    #<-9>,R1,R1             ;MODULO 512
                     50    34 A3   3C  09D1  1967          MOVZWL  CRB$L_INTD+VEC$W_MAPREG(R3),R0 ;MOVE START MAP REG. NO.
                        51   50    C0  09D5  1968          ADDL    R0,R1                   ;R1 HAS FINAL MAP REG. NO.
                                  09D8  1969                                  ;**R2 HAS START ADDR. OF 1ST MAP REGISTER
                                  09D8  1970                                  ;**R3 HAS CRB ADDRESS
                                  09D8  1971                                  ;** WHICH WERE GOTTEN FROM IOC$PURGDATAP
                   51   01EF 8F   B1  09D8  1972          CMPW    #495,R1                 ;LEGAL MAP REG. NO.?
                             1A   14  09DD  1973          BGTR    25$                     ;BR IF YES
                   51   01EF 8F   3C  09DF  1974          MOVZWL  #495,R1                 ;MAKE IT FINAL MAP REG. NO.=495
                00CC C5    6241   D0  09E4  1975          MOVL    (R2)[R1],UCB$L_MS_FMPR(R5) ;SAVE FINAL MAP REG
            00D0 C5    FC A241   D0  09EA  1976          MOVL    -4(R2)[R1],UCB$L_MS_PMPR(R5) ;SAVE PREVIOS MAP REG
                00D4 C5    6241   D0  09F1  1977          MOVL    (R2)[R1],UCB$L_MS_NMPR(R5) ;SAVE NEXT MAP REG(=495)
                             1E   11  09F7  1978          BRB     27$                     ;GO AHEAD NOW
                                  09F9  1979  25$:                            ;LEGAL MAP REG. NO.
                00CC C5    6241   D0  09F9  1980          MOVL    (R2)[R1],UCB$L_MS_FMPR(R5) ;SAVE FINAL MAP REG
                  00D0 C5    01   CE  09FF  1981          MNEGL   #1,UCB$L_MS_PMPR(R5)    ;ASSUME NO PREVIOUS MAP REG
                        51   50   D1  0A04  1982          CMPL    R0,R1                   ;WAS FINAL MAP REG. 1ST MAP REG
                             07   13  0A07  1983          BEQL    26$                     ;YES, THERE WAS NO PREVIOUS ONE
            00D0 C5    FC A241   D0  0A09  1984          MOVL    -4(R2)[R1],UCB$L_MS_PMPR(R5) ;OTHERWISE,SAVE PREVIOUS MAP REG
                                  0A10  1985  26$:
            00D4 C5    04 A241   D0  0A10  1986          MOVL    4(R2)[R1],UCB$L_MS_NMPR(R5) ;SAVE NEXT MAP REG
                                  0A17  1987  27$:
                    00C6 C5    95   0A17  1988          TSTB    UCB$B_MS_DPN(R5) ;TEST IF DIRECT DATAPATH?
                         0A    13   0A1B  1989          BEQL    30$                     ;YES
                                  0A1D  1990          ;*RELEASE DATAPATH AND MAP REG.'S
```

```
                                0A1D  1991          RELDPR                          ;RELEASE DATA PATH
            68 A5   20   AA     0A23  1992          BICW    #UCB$M_MS_RDPR,UCB$W_DEVSTS(R5) ;CLEAR FLAG
                                0A27  1993 30$:                                   ;
                                0A27  1994          RELMPR                          ;RELEASE MAP REGISTERS
                                0A2D  1995
                          0D    11    0A2D  1996          BRB     XTC1                    ;
                                0A2F  1997
                                0A2F  1998
                                0A2F  1999 ;
                                0A2F  2000 ; XTC - PROCESS TERMINATION CODE
                                0A2F  2001 ; HERE, THE FINAL STATUS CODE IS PUT IN R0, &
                                0A2F  2002 ; THE DRIVE STATUS IS RECORDED INTO UCB
                                0A2F  2003 ;
                                0A2F  2004
                                0A2F  2005 XTC:
                00C4 C5   B5    0A2F  2006          TSTW    UCB$W_MS_XC(R5)         ;SHOULD ANYTHING XFERRED?
                          07    13    0A33  2007          BEQL    XTC1                    ;NO, BRANCH
      00C4 C5   00FC C5   A2    0A35  2008          SUBW    UCB$W_MS_RBPC(R5),UCB$W_MS_XC(R5) ;GET ACTUALLY XFERRED
                                0A3C  2009 XTC1:
                          07    E1    0A3C  2010          BBC     #MS_XSR0_V_MOT,-        ;DID TAPE MOVE?
                04 00FE C5          0A3E  2011                  UCB$W_MS_XSR0(R5),7$            ;BR IF NO
                46 A5   07    AA    0A42  2012          BICW    #<MT$M_BOT!-            ;CLEAR BOT, AND
                                0A46  2013                  MT$M_EOF!-              ;END OF FILE,
                                0A46  2014                  MT$M_EOT>@-16,UCB$L_DEVDEPEND+2(R5) ;END OF TAPE
                                0A46  2015 7$:
                03    01    EF    0A46  2016          EXTZV   #MS_TSSR_V_TCC,#MS_TSSR_S_TCC,- ;EXTRACT TERMINATION CODE
             50   00C2 C5          0A49  2017                  UCB$W_MS_TSSR(R5),R0    ; INTO R0
                                0A4D  2018          CASE    R0,<-                   ;DISPATCH TO ROUTINES
                                0A4D  2019                  100$,-                  ;NORMAL TERMINATION
                                0A4D  2020                  110$,-                  ;ATTENTION CONDITION
                                0A4D  2021                  120$,-                  ;TAPE STATUS SLERT
                                0A4D  2022                  130$,-                  ;FUNCTION REJECT
                                0A4D  2023                  140$,-                  ;RECOVERABLE ERROR(TAPE MOVED)
                                0A4D  2024                  150$,-                  ;RECOVERABLE ERROR(TAPE NOT MOVED)
                                0A4D  2025                  160$,-                  ;UNRECOVERABLE ERROR(TAPE POSI LOST)
                                0A4D  2026                  170$,-                  ;FATAL CONTROLLER ERROR
                                0A4D  2027                  >
                                0A61  2028 ;
                                0A61  2029 ; FATLAL CONTROLLER ERROR(TCC=7)
                                0A61  2030 ;
                                0A61  2031
                                0A61  2032 170$:                                  ;
             50   0054 8F    3C    0A61  2033          MOVZWL  #SS$_CTRLERR,R0         ;PUT IN FINAL STATUS CODE
                    FD16    31    0A66  2034          BRW     FATALERR                ;GOTO FATAL ERROR
                                0A69  2035
                                0A69  2036 ;
                                0A69  2037 ; UNRECOVERABLE ERROR(TAPE POSITION LOST)
                                0A69  2038 ; (TCC=6)
                                0A69  2039
                                0A69  2040 160$:
               46 A5    10    A8    0A69  2041          BISW    #<MT$M_LOST@-16>,UCB$L_DEVDEPEND+2(R5) ;MARK POSITION LOST
             50   008C 8F    3C    0A6D  2042          MOVZWL  #SS$_DRVERR,R0          ;PUT IN FINAL STATUS CODE
                          06    E0    0A72  2043          BBS     #MS_XSR0_V_ONL,-        ;CHECK IF ON-LINE
                05 00FE C5          0A74  2044                  UCB$W_MS_XSR0(R5),165$  ;BR IF YES
             50   01A4 8F    3C    0A78  2045          MOVZWL  #SS$_MEDOFL,R0          ;NO, RETURN MEDIUM OFFLINE
                                0A7D  2046 165$:
                    FCFF    31    0A7D  2047          BRW     FATALERR                ;
```

```
                        0A80  2048
                        0A80  2049 ;
                        0A80  2050 ; NORMAL TERMINATION
                        0A80  2051 ; (TCC=0)
                        0A80  2052
                        0A80  2053 100$:                                        ;
      50   01    3C     0A80  2054          MOVZWL  #SS$_NORMAL,R0              ;PUT IN STATUS CODE
                        0A83  2055 101$:
009C C5    02    C0     0A83  2056          ADDL    #2,UCB$L_DPC(R5)            ;ADJUST TO CORRECT RETURN ADDRESS
     009C D5    17     0A88  2057          JMP     @UCB$L_DPC(R5)              ;RETURN TO DRIVER
                        0A8C  2058
                        0A8C  2059 ;
                        0A8C  2060 ; ATTENTION CONDITION
                        0A8C  2061 ; DRIVE HAS UNDERGONE STATUS CHANGE SUCH AS GOING OFFLINE OR COMING ONLINE
                        0A8C  2062 ; (TCC=1)
                        0A8C  2063
                        0A8C  2064 110$:
          06    E1      0A8C  2065          BBC     #MS_XSR0_V_ONL,-            ;CHECK IF ONLINE?
  06 00FE C5              0A8E  2066                  UCB$W_MS_XSR0(R5),112$      ;BR IF OFFLINE.
      50   05    3C     0A92  2067          MOVZWL  #TCC_REN,R0                 ;BECOME ONLINE, BUT
                        0A95  2068                                              ;SHOULDN'T HAVE BEEN OFFLINE
           00B3   31     0A95  2069          BRW     150$                        ;RETRY THE COMMAND
                        0A98  2070 112$:
50    0093 C5    90     0A98  2071          MOVB    UCB$B_CEX(R5),R0            ; GET HARDWARE COMMAND INDEX
      50   01    91     0A9D  2072          CMPB    #CDHC_UNL,R0                ;WAS IT UNLOAD?
           DE    13     0AA0  2073          BEQL    100$                        ;YES, ITS OK
50    01A4 8F    3C     0AA2  2074          MOVZWL  #SS$_MEDOFL,R0              ;MARK AS MEDIUM OFFLINE
           FCD5   31     0AA7  2075          BRW     FATALERR                    ;GOTO FATAL ERROR
                        0AAA  2076
                        0AAA  2077 ;
                        0AAA  2078 ; TAPE STATUS ALERT
                        0AAA  2079 ;(BITS OF INTEREST: TMK, LET, RLS, EOT, RIB, AND RLL)
                        0AAA  2080 ;**LET BIT IS LOGICAL END OF TAPE FOR DOS, NOT USED FOR NOW**
                        0AAA  2081 ; (TCC=2)
                        0AAA  2082
                        0AAA  2083 ; The reverse into BOT must return the status SS$_NORMAL becasue
                        0AAA  2084 ; at this time BACKUP depends on this fact and that is how the
                        0AAA  2085 ; other tape drivers work. This has been modified again. So that the
                        0AAA  2086 ; read reverse which BAKCUP doesn't depend on returns SS$_ENDOFFILE
                        0AAA  2087 ; when it encounters the BOT marker.
                        0AAA  2088
                        0AAA  2089 120$:                                        ;
          00    E1      0AAA  2090          BBC     #MS_XSR3_V_RIB,-            ;REVERSE INTO BOT?
  16 0104 C5              0AAC  2091                  UCB$W_MS_XSR3(R5),121$      ;
  46 A5    01    A8     0AB0  2092          BISW    #<MT$M_BOT@-16>,UCB$L_DEVDEPEND+2(R5) ;YES
     00B0  C5    D4     0AB4  2093          CLRL    UCB$L_RECORD(R5)
0092 C5    0F    91     0AB8  2094          CMPB    #CDHC_RDP,UCB$B_FEX(R5) ; Is this a read reverse?
           C1    12     0ABD  2095          BNEQ    100$                        ; If NEQ then return NORMAL code
50    0870 8F    3C     0ABF  2096          MOVZWL  #SS$_ENDOFFILE,R0           ; Take error return.
           BD    11     0AC4  2097          BRB     101$
                        0AC6  2098
                        0AC6  2099 121$:
          0C    E1      0AC6  2100          BBC     #MS_XSR0_V_RLL,-            ;CHECK IF RECORD LENGTH LONG?
  07 00FE C5              0AC8  2101                  UCB$W_MS_XSR0(R5),122$      ;TAKE NORMAL RETURN, IF NOT
50    0838 8F    3C     0ACC  2102          MOVZWL  #SS$_DATAOVERUN,R0          ;YES, ITS DATAOVERRUN
           B0    11     0AD1  2103          BRB     101$                        ;TAKE NORMAL RETURN
                        0AD3  2104 122$:
```

```
                     0F   E1   0AD3  2105          BBC     #MS_XSRO_V_TMK,-          ;CHECK IF SEE TAPE MARK
          21 00FE    C5        0AD5  2106                  UCB$W_MS_XSRO(R5),125$    ; ??
          46 A5      02   A8   0AD9  2107          BISW    #<MT$M_EOF@-16>,UCB$L_DEVDEPEND+2(R5) ;YES
        0092 C5      16   91   0ADD  2108          CMPB    #CDHC_WTM,UCB$B_FEX(R5)   ;WAS IT WRITE TMK?
                     16   13   0AE2  2109          BEQL    125$                      ;YES, LOOK FOR EOT
        0092 C5      05   91   0AE4  2110          CMPB    #CDHC_STR,UCB$B_FEX(R5)   ;WAS IT SKIPFILE REVERSE?
                     21   13   0AE9  2111          BEQL    128$                      ;YES
        0092 C5      02   91   0AEB  2112          CMPB    #CDHC_STF,UCB$B_FEX(R5)   ;WAS IT SKIPFILE FORWARD?
                     1A   13   0AF0  2113          BEQL    128$                      ;YES
                          0AF2  2114                                  ;**NOTE UCB$L_RECORD WAS ADJUSTED**
        50   0870    8F   3C   0AF3  2115          MOVZWL  #SS$_ENDOFFILE,R0         ;SET EOF
                FF89     31   0AF7  2116          BRW     101$                      ;TAKE NORMAL RETURN
                          0AFA  2117  125$:
                     00   E1   0AFA  2118          BBC     #MS_XSRO_V_EOT,-          ;CHECK IF AT EOT?
          0C 00FE    C5        0AFC  2119                  UCB$W_MS_XSRO(R5),128$    ;
          46 A5      04   A8   0B00  2120          BISW    #<MT$M_EOT@-16>,UCB$L_DEVDEPEND+2(R5) ;YES, SET FLAG
        50   0878    8F   3C   0B04  2121          MOVZWL  #SS$_ENDOFTAPE,R0         ;WRITE ERROR INTO EOT
                FF77     31   0B09  2122          BRW     101$                      ;
                          0B0C  2123  128$:                                       ;ANYTHING ELSE?
                FF71     31   0B0C  2124          BRW     100$                      ;TAKE NORMAL RETURN**TEMP**
                          0B0F  2125
                          0B0F  2126  ;
                          0B0F  2127  ; FUNCTION REJECT
                          0B0F  2128  ;(BITS OF INTEREST:BOT,WLK,VCK,ONL,ILA,ILC,NEF,WLE)
                          0B0F  2129  ; (TCC=3)
                          0B0F  2130
                          0B0F  2131  130$:
        50   008C    8F   3C   0B0F  2132          MOVZWL  #SS$_DRVERR,R0            ;MARK AS DRIVE ERROR
                     01   E1   0B14  2133          BBC     #MS_XSRO_V_BOT,-          ;CHECK IF AT BOT
          08 00FE    C5        0B16  2134                  UCB$W_MS_XSRO(R5),132$    ;
          46 A5      01   A8   0B1A  2135          BISW    #<MT$M_BOT@-16>,UCB$L_DEVDEPEND+2(R5) ;YES
             0080    C5   D4   0B1E  2136          CLRL    UCB$L_RECORD(R5)
                          0B22  2137  132$:                                      ;
                     04   E1   0B22  2138          BBC     #MS_XSRO_V_VCK,-          ;WAS VOLUME CHECK?
          06 00FE    C5        0B24  2139                  UCB$W_MS_XSRO(R5),134$    ;
        68 A5      1000 8F   A8   0B28  2140          BISW    #UCB$M_MS_VCK,UCB$W_DEVSTS(R5) ;YES,RECORD IT
                          0B2E  2141  134$:
                     08   E1   0B2E  2142          BBC     #MS_XSRO_V_WLE,-          ;CHECK IF WRITE LOCK ERROR
          09 00FE    C5        0B30  2143                  UCB$W_MS_XSRO(R5),136$    ;
          46 A5      08   A8   0B34  2144          BISW    #<MT$M_WLK@-16>,UCB$L_DEVDEPEND+2(R5) ;YES, SET FLAG
        50   025C    8F   3C   0B38  2145          MOVZWL  #SS$_WRITLCK,R0           ;MARK AS WRITE-LOCKED ERROR
                          0B3D  2146  136$:                                       ;
                     06   E0   0B3D  2147          BBS     #MS_XSRO_V_ONL,-          ;CHECK IF ONLINE
          05 00FE    C5        0B3F  2148                  UCB$W_MS_XSRO(R5),138$    ;BR IF YES
        50   01A4    8F   3C   0B43  2149          MOVZWL  #SS$_MEDOFL,R0            ;MARK MEDIUM OFFLINE
                          0B48  2150  138$:                                       ;
                FC34     31   0B48  2151          BRW     FATALERR                  ;TAKE FATAL OR HARD ERROR RETURN
                          0B4B  2152
                          0B4B  2153  ;
                          0B4B  2154  ; RECOVERABLE ERROR(TAPE MOVED)
                          0B4B  2155  ; RECOVERABLE ERROR(TAPE NOT MOVED)
                          0B4B  2156  ;(TCC=4 OR 5)
                          0B4B  2157
                          0B4B  2158  140$:                                       ;
                          0B4B  2159
                          0B4B  2160  150$:                                       ;
      13 009A C5      0F   E0   0B4B  2161          BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),155$ ;IF SET, RETRY INHIBITED
```

```
  7E   009C D5  32 0B51 2162        CVTWL   @UCB$L_DPC(R5),-(SP)    ;GET BRANCH DISPLACEMENT
009C C5     8E  C0 0B56 2163        ADDL    (SP)+,UCB$L_DPC(R5)     ;CALCULATE RETURN ADDRESS -2
009C C5     02  C0 0B5B 2164        ADDL    #2,UCB$L_DPC(R5)        ;ADUST TO CORRECT RETURN ADDRESS
       009C D5  17 0B60 2165        JMP     @UCB$L_DPC(R5)          ;RETURN TO DRIVER
                   0B64 2166
          FEFA  31 0B64 2167 155$:  BRW     170$                    ;RETURN AS FATAL
                   0B67 2168
```

```
                          0B67  .SBTTL  TS11/TS04 INTERRUPT SERVICE ROUTINE
                          0B67  ;+
                          0B67  ; TS$INT - TS11/TS04 MAGTAPE INTERRUPTS
                          0B67  ;
                          0B67  ; THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT OCCURS
                          0B67  ; ON TS11/TS04 CONTROLLER. THE STATE OF THE STACK ON ENTRY IS:
                          0B67  ;
                          0B67  ;       00(SP) = ADDR. OF IDB ADDRESS
                          0B67  ;       04-28(SP) = SAVED R0-R5
                          0B67  ;       32(SP) = INTERRUPT PC
                          0B67  ;       36(SP) = INTERRUPT PSL
                          0B67  ;
                          0B67  ; INTERRUPT DISPATCHING OCCURS AS FOLLOWS:
                          0B67  ;
                          0B67  ;(MUMBLE)
                          0B67  ;
                          0B67  ;-
                          0B67
                          0B67  TS$INT::
          53   9E   D0    0B67          MOVL    @(SP)+,R3                       ;GET ADDR. OF IDB
          54   63   7D    0B6A          MOVQ    IDB$L_CSR(R3),R4                ;GET CONTROLLER CSR AND UCB ADDR.
     50 00B6   C5   D0    0B6D          MOVL    UCB$L_MS_TSPT1(R5),R0           ;COMMAND PACKET ADDR. IN R0
     00C0 C5   64   B0    0B72          MOVW    (R4),UCB$W_MS_TSBA(R5)          ;GET DEVICE REGISTER TSBA(TSDB)
 00C2 C5   02  A4   B0    0B77          MOVW    2(R4),UCB$W_MS_TSSR(R5)         ;GET TSSR INTO UCB
   23 64 A5    01   E5    0B7D          BBCC    #UCB$V_INT,UCB$W_STS(R5),10$ ;IF CLR, INTERRUPT NOT EXPECTED
 00F8 C5   10  A0   B0    0B82          MOVW    MS_MHD(R0),UCB$W_MS_MHD(R5)  ;SAVE MSG BUFFER IN UCB
 00FA C5   12  A0   D0    0B88          MOVL    MS_LNH(R0),UCB$W_MS_LNH(R5)  ;SAVE NEXT LONG WORD
 00FE C5   16  A0   7D    0B8E          MOVQ    MS_XSR0(R0),UCB$Q_MS_XSR0(R5) ;SAVE REST OF MSG BUFFER
      53   10  A5   D0    0B94          MOVL    UCB$L_FR3(R5),R3                ;RESTORE REMAINING DRIVER CONTEXT
           0C  B5   16    0B98          JSB     @UCB$L_FPC(R5)                  ;CALL DRIVER
                          0B9B  5$:
          50   8E   7D    0B9B          MOVQ    (SP)+,R0                        ;RESTORE REGISTERS
          52   8E   7D    0B9E          MOVQ    (SP)+,R2                        ;
          54   8E   7D    0BA1          MOVQ    (SP)+,R4                        ;
                    02    0BA4          REI                                     ;RETURN FROM INTERRUPT
                          0BA5
                          0BA5  ;
                          0BA5  ; NON-QIO RESPONSE INTERRUPT
                          0BA5  ;
                          0BA5
                          0BA5  10$:
   02 68 A5    0A   E4    0BA5          BBSC    #UCB$V_MS_LBA,UCB$W_DEVSTS(R5),20$ ;YES. LOADING BUFFER ADDR.?
               EF   11    0BAA          BRB     5$                              ; Branch to dismiss interrupt.
                          0BAC
                          0BAC  ;
                          0BAC  ; HERE, WAS LOADING BUFFER ADDRESS
                          0BAC  ;
                          0BAC  20$:
               0A   E0    0BAC          BBS     #MS_TSSR_V_NBA,-
   E9 00C2 C5             0BAE                  UCB$W_MS_TSSR(R5),5$    ;FAIL TO LOAD BUFFER ADDR.
                          0BB2
                          0BB2  ;
                          0BB2  ; BUFFER ADDRESS LOADED SUCESSFULLY
                          0BB2  ; DO RELEASE MESSAGE BUFFER TO TS11/TS04
                          0BB2  ;
                          0BB2  30$:
                          0BB2                                                  ;**MUST BE AT BOT
      46 A5    01   A8    0BB2          BISW    #<MTSM_BOT@-16>,UCB$L_DEVDEPEND+2(R5) ;MARK IT
```

```
                 1A 64 A5    05    E1   0BB6  2227          BBC     #UCB$V_POWER,UCB$W_STS(R5),35$ ;BR IF NOT POWERFAIL
                 15 68 A5    0B    E0   0BBB  2228          BBS     #UCB$V_MS_RPI,UCB$W_DEVSTS(R5),35$ ;BR IF REPOSITION IN PROGRESS
                 09 68 A5    06    E1   0BC0  2229          BBC     #UCB$V_MS_SWE,UCB$W_DEVSTS(R5),34$ ;BR IF NOT SOFTWARE EMULATION
        00F4 C5    00D0 C5          D0   0BC5  2230          MOVL    UCB$L_MS_PMPR(R5),UCB$L_MS_TPOSITN(R5) ;GET FROM ELSEWHERE
                           07        11   0BCC  2231          BRB     35$
                                           0BCE  2232  34$:
        00F4 C5    00B0 C5          D0   0BCE  2233          MOVL    UCB$L_RECORD(R5),UCB$L_MS_TPOSITN(R5) ;SAVE TAPE POSITION
                                           0BD5  2234  35$:
                   00B0 C5          D4   0BD5  2235          CLRL    UCB$L_RECORD(R5)                ;
                                           0BD9  2236                                               ; WHICH TELL TAPE POSITION
                      FFBF          31   0BD9  2237          BRW     5$                              ;DO RETURN FROM INTERRUPT
                                           0BDC  2238
                                           0BDC  2239
```

```
                        0BDC  2241 .SBTTL  TIMEOUT HANDLER
                        0BDC  2242 ;+MSTMO - HANDLES TIME-OUT WHEN TS11/TS04 DOES NOT INTERRUPT AFTER
                        0BDC  2243 ; A HARDWARE COMMAND ISSUED FOR A SPECIFIED PERIOD OF TIME.
                        0BDC  2244 ; THE ROUTINE DEALLOCATES DATA PATH AND MAP REGISTER IF IT'S DATA
                        0BDC  2245 ; TRANSFER COMMAND, AND ABORTS THE I/O OPERATION.
                        0BDC  2246 ; IF IT WAS DUE TO POWERFAIL, REPOSITIONING IS ATTEMPTED, AND
                        0BDC  2247 ; THE TIME-OUTED IRP IS RE-ISSUED
                        0BDC  2248 ;
                        0BDC  2249 ; INPUT:
                        0BDC  2250 ;
                        0BDC  2251 ; OUTPUT:
                        0BDC  2252
                        0BDC  2253                      .ENABL  LSB                             ;ENABLE LOCAL SYMBOL
                        0BDC  2254 MSTMO:
                        0BDC  2255                      SETIPL  UCB$B_FIPL(R5)                  ;LOWER IPL TO DEVICE FORK LEVEL
                        0BE0  2256                                      ;**ASSUME NO PURGING OF DATAPATH FOR TIMEOUT**
06 68 A5   05   E5      0BE0  2257                      BBCC    #UCB$V_MS_RDPR,UCB$W_DEVSTS(R5),1$ ;BR IF DATAPATH NOT REQUESTED
                        0BE5  2258                      RELDPR                                  ;RELEASE DATA PATH
                        0BEB  2259 1$:
                        0BEB  2260                      RELMPR                                  ;RELEASE MAP REGISTERS
                   04   11  0BF1  2261                      BRB 2$                                  ;
                        0BF3  2262 ;
                        0BF3  2263 ; TIMEOUT FOR NON-I/O XFR OPERATION
                        0BF3  2264 ;
                        0BF3  2265
                        0BF3  2266 MSTMO1:
                        0BF3  2267                      SETIPL  UCB$B_FIPL(R5)                  ;LOWER IPL TO DEVICE FORK LEVEL
                        0BF7  2268 2$:
                     05 E4  0BF7  2269                      BBSC    #UCB$V_POWER,-                  ; Branch around to reposition if
                  03 64 A5        0BF9  2270                              UCB$W_STS(R5),5$                ;  we had POWERFAIL.
                     00CF 31  0BFC  2271                      BRW     90$
                        0BFF  2272 ;
                        0BFF  2273 ; HERE, CHECK DRIVE OFF-LINE UNLOADED OR NOT
                        0BFF  2274 ;
                        0BFF  2275
                        0BFF  2276 5$:
                     F5FE 30  0BFF  2277                      BSBW    TEST_NBA                        ; Test to assure we DON'T need TS11
                        0C02  2278                                                              ;  message buffer address loaded.
                     03 50 E8  0C02  2279                      BLBS    R0,6$                           ; LBS implies TS11 READY and able.
                      FB77 31  0C05  2280                      BRW     FATALERR                        ; If TS11 not ready, we can't even
                        0C08  2281                                                              ;  try to reposition.
                        0C08  2282 6$:
          00000000'GF   16  0C08  2283                      JSB     G^EXE$READ_TODR                 ;GET CURRENT TIME OF DAY
50        000005DC 8F   C0  0C0E  2284                      ADDL    #1500,R0                        ;ADD 15 SEC. TO WAIT
           7C A5    50   D0  0C15  2285                      MOVL    R0,UCB$W_BOFF(R5)               ;STORE IT IN UCB
                        0C19  2286 ;
                        0C19  2287 ; HERE, GET TS11'S CSR EQUIVALENT INTO R4
                        0C19  2288 ;
                        0C19  2289
                        0C19  2290 7$:
                        0C19  2291                      DSBINT                                  ;DISABLE INTERRUPTS
                        0C1F  2292                      WFIKPCH 8$,#2                           ;WAITFOR INTERRUPT OR TIMEOUT
                        0C29  2293                      IOFORK                                  ;
                        0C2F  2294 8$:
                        0C2F  2295                      SETIPL  UCB$B_FIPL(R5)                  ;LOWER IPL TO FORK LEVEL
                        0C33  2296                      EXHC    FAIL,RC_RWD                     ;DO A REWIND
                     0020 31  0C38  2297                      BRW     9$                              ;BR IF SUCCESS=>DRIVE ONLINE
```

```
                                0C3E  2298 FAIL:
                                0C3E  2299 ;
                                0C3E  2300 ; HERE, TO SEND MESSAGE TO OPERATOR TO INFORM DRIVE OFFLINE
                                0C3E  2301 ;
                                0C3E  2302
        00000000'GF   16        0C3E  2303          JSB     G^EXE$READ_TODR             ;GET CURRENT TIME OF DAY
         7C A5    50   D1        0C44  2304          CMPL    R0,UCB$W_BOFF(R5)           ;15 SEC. PASSED?
                  CF   1B        0C48  2305          BLEQU   7$                          ;NO , GO TRY AGAIN
                                0C4A  2306
                                0C4A  2307
          54   00'8F   9A        0C4A  2308          MOVZBL  #MSG$_DEVOFFLIN,R4          ;SET MESSAGE NUMBER
     53   00000000'GF   9E        0C4E  2309          MOVAB   G^SYS$GL_OPRMBX,R3          ;GET ADDRESS OF OPERATOR MAILBOX
        00000000'GF   16        0C55  2310          JSB     G^EXE$SNDEVMSG              ;SEND MESSAGE TO OPERATOR
                FFAA   31        0C5B  2311          BRW     6$                          ;
                                0C5E  2312 ;
                                0C5E  2313
                                0C5E  2314 ; OTHERWISE DO REPOSITIONING TAPE
                                0C5E  2315 ;
                                0C5E  2316 9$:
         68 A5    0800 8F   A8        0C5E  2317          BISW    #UCB$M_MS_RPI,UCB$W_DEVSTS(R5) ;FLAG REPOSITION IN PROGRESS
                                0C64  2318          EXHC    50$,HC_RWD                  ;DO REWIND 1ST
                                0C6C  2319                                              ; & CLEAR UCB$L_RECORD
                                0C6C  2320 10$:
      00F4 C5    00B0 C5   D1        0C6C  2321          CMPL    UCB$L_RECORD(R5),UCB$L_MS_TPOSITN(R5) ;CHECK REPOSITIONING
                     46   13        0C73  2322          BEQL    80$                         ;BR IF YES
      00F4 C5    00B0 C5   D1        0C75  2323          CMPL    UCB$L_RECORD(R5),UCB$L_MS_TPOSITN(R5) ;IS IT GTR THAN
                     33   14        0C7C  2324          BGTR    50$                         ;BR IF YES
   50 00F4 C5    00B0 C5   C3        0C7E  2325          SUBL3   UCB$L_RECORD(R5),UCB$L_MS_TPOSITN(R5),R0 ;GET WHAT'S LEFT
    50   00007FFF 8F   D1        0C86  2326          CMPL    #^X7FFF,R0                  ;LESS THAN 32,768?
                     09   14        0C8D  2327          BGTR    20$                         ;BR IF YES
      00B4 C5    7FFF 8F   B0        0C8F  2328          MOVW    #^X7FFF,UCB$W_MS_SPACNT(R5) ;SKIP 32,768 RECORDS TILL DONE
                     05   11        0C96  2329          BRB     30$                         ;
                                0C98  2330 20$:
        00B4 C5    50   B0        0C98  2331          MOVW    R0,UCB$W_MS_SPACNT(R5)  ;SKIP WHAT'S LEFT
                                0C9D  2332 30$:
                                0C9D  2333          EXHC    50$,HC_SRF                  ;
        51   00C4 C5    3C        0CA5  2334          MOVZWL  UCB$W_MS_XC(R5),R1          ;GET NO. OF RECORDS PASSED
        00B0 C5    51   C0        0CAA  2335          ADDL    R1,UCB$L_RECORD(R5)         ;UPDATE TAPE POSITION
                     BB   11        0CAF  2336          BRB     10$                         ;GO BACK
                                0CB1  2337 50$:
    68 A5    0800 8F   AA        0CB1  2338          BICW    #UCB$M_MS_RPI,UCB$W_DEVSTS(R5) ;CLEAR FLAG, REPOSI. FAILED
             FABF CF   17        0CB7  2339          JMP     FATALERO                    ;
                                0CBB  2340 ;
                                0CBB  2341 ; HERE, GO AHEAD WITH THE CURRENT QIO
                                0CBB  2342 ;
                                0CBB  2343 80$:
    68 A5    0800 8F   AA        0CBB  2344          BICW    #UCB$M_MS_RPI,UCB$W_DEVSTS(R5) ;CLEAR FLAG,REPOSITION DONE
         53   58 A5    D0        0CC1  2345          MOVL    UCB$L_IRP(R5),R3            ;R3 HAS IRP ADDRESS
      78 A5   2C A3    7D        0CC5  2346          MOVQ    IRP$L_SVAPTE(R3),UCB$L_SVAPTE(R5) ;RESTORE XFER PARAMETERS
             F5F3 CF   17        0CCA  2347          JMP     TS_STARTIO
                                0CCE  2348 90$:
         00000000'GF   16        0CCE  2349          JSB     G^ERL$DEVICTMO              ;LOG TIMEOUT ERROR
        50   022C 8F   3C        0CD4  2350          MOVZWL  #SS$_TIMEOUT,R0             ;SET TIMEOUT STATUS
                                0CD9  2351          .IF     DF      TS_TRACE
                                0CD9  2352          BSBW    TRACE_STATUS                ; Trace final I/O status.
                                0CD9  2353          .ENDC
                                0CD9  2354          REQCOM                              ;GO COMPLETE I/O REQUEST PROCESSING
```

```
                                                M 6
                        OCDF  2355
                        OCDF  2356              .DSABL  LSB
```

```
                     0CDF  2358 .SBTTL  TS11/TS04 REGISTER DUMP ROUTINE
                     0CDF  2359 ;+
                     0CDF  2360 ; TS_REGDUMP - TS11/TS04 REGISTER DUMP ROUTINE
                     0CDF  2361 ;
                     0CDF  2362 ; THIS ROUTINE IS CALLED TO SAVE THE CONTROLLER AND DRIVE REGISTERS IN A
                     0CDF  2363 ; SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERRORLOGGING ROUTINE AND
                     0CDF  2364 ; FROM THE DIAGNOSTIC BUFFER FILL ROUTINE
                     0CDF  2365 ;
                     0CDF  2366 ; INPUT:
                     0CDF  2367 ;       R0 = ADDRESS OF REGISTER SAVE BUFFER
                     0CDF  2368 ;       R4 = ADDRESS OF CSR (EQUIVALENT)
                     0CDF  2369 ;       R5 = UCB ADDRESS
                     0CDF  2370 ;
                     0CDF  2371 ; OUTPUT:
                     0CDF  2372 ;
                     0CDF  2373 ; THE CONTROLLER AND DRIVE REGISTERS ARE SAVED IN THE SPECIFIED BUFFER
                     0CDF  2374 ;-
                     0CDF  2375
                     0CDF  2376 TS_REGDUMP:
     80   17   D0    0CDF  2377         MOVL    #23,(R0)+                       ;23 REGISTERS FOLLOW TO BE DUMPED
80   00C0 C5   3C    0CE2  2378         MOVZWL  UCB$W_MS_TSBA(R5),(R0)+         ;GET TSBA
80   00C2 C5   3C    0CE7  2379         MOVZWL  UCB$W_MS_TSSR(R5),(R0)+         ;GET TSSR
80   00C6 C5   9A    0CEC  2380         MOVZBL  UCB$B_MS_DPN(R5),(R0)+          ;GET DATAPATH NO.
80   00C8 C5   D0    0CF1  2381         MOVL    UCB$L_MS_DPR(R5),(R0)+          ;GET DATAPATH REG.
80   00CC C5   D0    0CF6  2382         MOVL    UCB$L_MS_FMPR(R5),(R0)+         ;GET FINAL MAP REGISTER
80   00D0 C5   D0    0CFB  2383         MOVL    UCB$L_MS_PMPR(R5),(R0)+         ;GET FINAL-1 MAP REGISTER
80   00D4 C5   D0    0D00  2384         MOVL    UCB$L_MS_NMPR(R5),(R0)+         ;GET FINAL+1 MAP REGISTER
51   00B6 C5   D0    0D05  2385         MOVL    UCB$L_MS_TSPT1(R5),R1           ;GET MESSAGE BUFFER ADDR
     52   0F   9A    0D0A  2386         MOVZBL  #15,R2                          ;15 WORDS IN MSG BUFFER
                     0D0D  2387 10$:
     80   81   3C    0D0D  2388         MOVZWL  (R1)+,(R0)+                     ;COPY FROM MSG BUFFER
                     0D10  2389                                 ;**FROM MS_CPHD TO MS_XSR3, SEE $DEFINI MS**
       FA 52   F5    0D10  2390         SOBGTR  R2,10$                          ;LOOP BACK
80   00C7 C5   9A    0D13  2391         MOVZBL  UCB$B_MS_PER(R5),(R0)+          ;GET PURGE ERROR INDICATOR
               05    0D18  2392         RSB                                     ;
                     0D19  2393
                     0D19  2394 TS_END:                                         ;ADDRESS OF LAST LOCATION IN DRIVER
                     0D19  2395
                     0D19  2396         .END
```

```
$$$                     = 00000020 R     02
$$GP                    = 00000002
ACP$ACLESS                ********   X   03
ACP$DEACCESS              ********   X   03
ACP$MODIFY                ********   X   03
ACP$MOUNT                 ********   X   03
ACP$READBLK               ********   X   03
ACP$WRITEBLK              ********   X   03
AT$_UBA                 = 00000001
CDHC_BRL                = 00000019
CDHC_CLN                = 00000018
CDHC_DRI                = 00000004
CDHC_ERS                = 00000006
CDHC_GST                = 00000015
CDHC_NOP                = 00000000
CDHC_PAK                = 00000008
CDHC_RDN                = 0000000C
CDHC_RDP                = 0000000F
CDHC_RPS                = 00000013
CDHC_RRN                = 00000010
CDHC_RRP                = 00000011
CDHC_RWD                = 00000003
CDHC_SCH                = 00000014
CDHC_SRF                = 00000009
CDHC_SRR                = 00000007
CDHC_STF                = 00000002
CDHC_STR                = 00000005
CDHC_UNL                = 00000001
CDHC_WCK                = 0000000A
CDHC_WDR                = 00000012
CDHC_WKR                = 0000000D
CDHC_WRC                = 0000001B
CDHC_WRD                = 0000000E
CDHC_WSM                = 0000001A
CDHC_WTM                = 00000016
CDHC_WTR                = 00000017
CLEAN                     00000711 R     03
CLS_MDE                 = 00000003
CLS_MDF                 = 00000001
CLS_ONF                 = 00000000
CLS_OTHER               = 00000001
CLS_PTB                 = 00000000
CLS_WLN                 = 00000002
CRB$L_INTD              = 00000024
DC$_TAPE                = 00000002
DDB$L_ACPD              = 00000010
DDB$L_DDT               = 0000000C
DEV$M_AVL               = 00040000
DEV$M_DIR               = 00000008
DEV$M_ELG               = 00400000
DEV$M_FOD               = 00004000
DEV$M_IDV               = 04000000
DEV$M_NNM               = 00000200
DEV$M_ODV               = 08000000
DEV$M_SDI               = 00000010
DEV$M_SQD               = 00000020
DEV$V_FOR               = 00000018
DEV$V_MNT               = 00000013
DPT$C_LENGTH            = 00000038
DPT$C_VERSION           = 00000004
DPT$INITAB                00000038 R     02
DPT$REINITAB              00000076 R     02
DPT$TAB                   00000000 R     02
DRVCLR                    0000071C R     03
DT$_TS11                = 00000004
DYN$C_CRB               = 00000005
DYN$C_DDB               = 00000006
DYN$C_DPT               = 0000001E
DYN$C_UCB               = 00000010
EMB$L_DV_REGSAV         = 0000004E
ERASE                     000006F0 R     03
ERL$DEVICERR              ********   X   03
ERL$DEVICTMO              ********   X   03
EXE$ALONONPAGED           ********   X   03
EXE$GL_NONPAGED           ********   X   03
EXE$IOFORK                ********   X   03
EXE$ONEPARM               ********   X   03
EXE$READ_TODR             ********   X   03
EXE$SETMODE               ********   X   03
EXE$SNDEVMSG              ********   X   03
EXE$ZEROPARM              ********   X   03
FAIL                      00000C3E R     03
FATALERO                  0000077A R     03
FATALERR                  0000077F R     03
FCC_CPE                 = 00000001
FCC_IDF                 = 00000000
FCC_LAP                 = 00000003
FCC_UPE                 = 00000002
FCNEXT                    00000795 R     03
FDISPATCH                 00000379 R     03
FUNCTAB_LEN             = 00000088
GETSTS                    00000727 R     03
HCEX                      000007EF R     03
HCTAB                     00000038 R     03
HC_BRL                  = 0000C08A
HC_CLN                  = 0000C28A
HC_DRI                  = 0000C08B
HC_ERS                  = 0000C189
HC_GST                  = 0000C08F
HC_NOP                  = 00000000
HC_PAK                  = 00000000
HC_RDN                  = 0000C081
HC_RDP                  = 0000C181
HC_RPS                  = 00000000
HC_RRN                  = 0000C381
HC_RRP                  = 0000C281
HC_RWD                  = 0000C488
HC_SCH                  = 00000000
HC_SRF                  = 0000C088
HC_SRR                  = 0000C188
HC_STF                  = 0000C088
HC_STR                  = 0000C188
HC_UNL                  = 0000C18A
HC_WCK                  = 00000000
```

```
HC_WDR                  = 0000C285
HC_WKR                  = 00000000
HC_WRC                  = 0000C084
HC_WRD                  = 0000C085
HC_WSM                  = 0000C086
HC_WTM                  = 0000C089
HC_WTR                  = 0000C289
IDB$L_CSR               = 00000000
IDB$L_OWNER             = 00000004
IO$M_NOWAIT             = 00000080
IO$V_INHRETRY           = 0000000F
IO$V_OPPOSITE           = 00000009
IO$V_REVERSE            = 00000006
IO$V_SWAP               = 00000008
IO$_ACCESS              = 00000032
IO$_ACPCONTROL          = 00000038
IO$_AVAILABLE           = 00000011
IO$_CLEAN               = 0000001E
IO$_CREATE              = 00000033
IO$_DEACCESS            = 00000034
IO$_DELETE              = 00000035
IO$_DRVCLR              = 00000004
IO$_ERASETAPE           = 00000006
IO$_MODIFY              = 00000036
IO$_MOUNT               = 00000039
IO$_NOP                 = 00000000
IO$_PACKACK             = 00000008
IO$_READLBLK            = 00000021
IO$_READPBLK            = 0000000C
IO$_READPRESET          = 00000019
IO$_READVBLK            = 00000031
IO$_RECAL               = 00000003
IO$_REREADN             = 00000016
IO$_REREADP             = 00000017
IO$_REWIND              = 00000024
IO$_REWINDOFF           = 00000022
IO$_SENSECHAR           = 0000001B
IO$_SENSEMODE           = 00000027
IO$_SETCHAR             = 0000001A
IO$_SETMODE             = 00000023
IO$_SKIPFILE            = 00000025
IO$_SKIPRECORD          = 00000026
IO$_SPACEFILE           = 00000002
IO$_SPACERECORD         = 00000009
IO$_UNLOAD              = 00000001
IO$_VIRTUAL             = 0000003F
IO$_WRITECHECK          = 0000000A
IO$_WRITELBLK           = 00000020
IO$_WRITEMARK           = 0000001C
IO$_WRITEOF             = 00000028
IO$_WRITEPBLK           = 0000000B
IO$_WRITERET            = 00000018
IO$_WRITEVBLK           = 00000030
IO$_WRTMKR              = 0000001D
IOC$CANCELIO            ******** X   03
IOC$DIAGBUFILL          ******** X   03
IOC$LOADUBAMAP          ******** X   03
IOC$LOADUBAMAPA         ******** X   03
IOC$MNTVER              ******** X   03
IOC$PURGDATAP           ******** X   03
IOC$RELDATAP            ******** X   03
IOC$RELMAPREG           ******** X   03
IOC$REQCOM              ******** X   03
IOC$REQDATAP            ******** X   03
IOC$REQMAPREG           ******** X   03
IOC$RETURN              ******** X   03
IOC$WFIKPCH             ******** X   03
IRP$L_MEDIA             = 00000038
IRP$L_SVAPTE            = 0000002C
IRP$L_WIND              = 00000018
IRP$S_FCODE             = 00000006
IRP$V_FCODE             = 00000000
IRP$V_PHYSIO            = 00000008
IRP$V_VIRTUAL           = 00000004
IRP$W_FUNC              = 00000020
IRP$W_STS               = 0000002A
LDTSDB                  000008EB R      03
MASKH                   = 00000008
MASKL                   = 04000000
MEDIA_ID_TS11           = 6CE9300B
MMG$GL_SPTBASE          ******** X   03
MS$DDT                  00000000 RG     03
MSG$_DEVOFFLIN          ******** X   03
MSGREL                  000006FB R      03
MSG_ATN                 = 00000013
MSG_END                 = 00000010
MSG_ERR                 = 00000012
MSG_FAL                 = 00000011
MSG_LOG                 = 00000014
MSTMO                   00000BDC R      03
MSTMO1                  00000BF3 R      03
MS_BA1                  00000004
MS_BACT                 00000002
MS_CHWD                 0000000E
MS_CNT                  00000006
MS_CPHD                 00000000
MS_CPHD_M_ACK           = 00008000
MS_CPHD_M_CVC           = 00004000
MS_CPHD_M_IE            = 00000080
MS_CPHD_M_OPP           = 00002000
MS_CPHD_M_SWB           = 00001000
MS_LNH                  00000012
MS_LNTH                 0000000C
MS_MBAO                 00000008
MS_MBA1                 0000000A
MS_MHD                  00000010
MS_RBPC                 00000014
MS_TSSR_S_TCC           = 00000003
MS_TSSR_V_NBA           = 0000000A
MS_TSSR_V_SSR           = 00000007
MS_TSSR_V_TCC           = 00000001
MS_XSRO                 00000016
MS_XSRO_V_BOT           = 00000001
MS_XSRO_V_EOT           = 00000000
```

```
MS_XSR0_V_MOT           =  00000007
MS_XSR0_V_ONL           =  00000006
MS_XSR0_V_RLL           =  0000000C
MS_XSR0_V_TMK           =  0000000F
MS_XSR0_V_VCK           =  00000004
MS_XSR0_V_WLE           =  0000000B
MS_XSR0_V_WLK           =  00000002
MS_XSR1                    00000018
MS_XSR2                    0000001A
MS_XSR3                    0000001C
MS_XSR3_V_RIB           =  00000000
MT$CHECK_ACCESS            ********    X   03
MT$K_NORMAL15           =  0000000E
MT$M_BOT                =  00010000
MT$M_EOF                =  00020000
MT$M_EOT                =  00040000
MT$M_HWL                =  00080000
MT$M_LOST               =  00100000
MT$S_FORMAT             =  00000004
MT$V_BOT                =  00000010
MT$V_EOF                =  00000011
MT$V_FORMAT             =  00000004
NOP                        000003F1  R       03
PACKACK                    000003EB  R       03
PMIS                       000008C9  R       03
PNOP                       00000861  R       03
PPOS                       0000086D  R       03
PR$_IPL                 =  00000012
PWCR                       000008CC  R       03
PWRFL1                     00000910  R       03
PXFR                       00000923  R       03
PXFRR                      0000092D  R       03
PXFRRD                     00000916  R       03
READDATA                   000003F9  R       03
READDATAR                  00000450  R       03
READPRESET                 000003F1  R       03
REREADN                    00000499  R       03
REREADP                    00000445  R       03
RET                        00000861  R       03
REWIND                     00000681  R       03
RFCNEXT                    00000795  R       03
SETCHAR                    000003D8  R       03
SIZ...                  =  00000001
SPCFILFOR                  00000512  R       03
SPCFILREV                  0000059F  R       03
SPCRECFOR                  000005F1  R       03
SPCRECREV                  00000667  R       03
SS$_CTRLERR             =  00000054
SS$_DATAOVERUN          =  00000838
SS$_DEVOFFLINE          =  00000084
SS$_DRVERR              =  0000008C
SS$_ENDOFFILE           =  00000870
SS$_ENDOFTAPE           =  00000878
SS$_ENDOFVOLUME         =  000009A0
SS$_MEDOFL              =  000001A4
SS$_NORMAL              =  00000001
SS$_TIMEOUT             =  0000022C
SS$_VOLINV              =  00000254
SS$_WRITLCK             =  0000025C
SYS$GL_OPRMBX              ********    X   03
TCC_ATN                 =  00000001
TCC_FNR                 =  00000003
TCC_FTL                 =  00000007
TCC_NML                 =  00000000
TCC_REM                 =  00000004
TCC_REN                 =  00000005
TCC_TSA                 =  00000002
TCC_UER                 =  00000006
TEST_NBA                   00000200  R       03
TS$INT                     00000B67  RG      03
TS_END                     00000D19  R       03
TS_FUNCTABLE               00000070  R       03
TS_INIT                    000000F8  R       03
TS_REGDUMP                 00000CDF  R       03
TS_STARTIO                 000002C1  R       03
UCB$B_CEX               =  00000093
UCB$B_DEVCLASS          =  00000040
UCB$B_DEVTYPE           =  00000041
UCB$B_DIPL              =  0000005E
UCB$B_ERTCNT            =  00000080
UCB$B_ERTMAX            =  00000081
UCB$B_FEX               =  00000092
UCB$B_FIPL              =  0000000B
UCB$B_MS_DPN               000000C6
UCB$B_MS_PER               000000C7
UCB$K_LCC_TAPE_LENGTH   =  000000B4
UCB$K_MS_LENGTH         =  00000106
UCB$L_CRB               =  00000024
UCB$L_DEVCHAR           =  00000038
UCB$L_DEVCHAR2          =  0000003C
UCB$L_DEVDEPEND         =  00000044
UCB$L_DPC               =  0000009C
UCB$L_FPC               =  0000000C
UCB$L_FR3               =  00000010
UCB$L_IOQFL             =  0000004C
UCB$L_IRP               =  00000058
UCB$L_MEDIA_ID          =  0000008C
UCB$L_MS_DPR               000000C8
UCB$L_MS_FMPR              000000CC
UCB$L_MS_NMPR              000000D4
UCB$L_MS_OMPR              000000D8
UCB$L_MS_PMPR              000000D0
UCB$L_MS_TIMOUT            000000DC
UCB$L_MS_TMP2              000000E8
UCB$L_MS_TPOSITN           000000F4
UCB$L_MS_TSPT1             000000B6
UCB$L_MS_TSPT2             000000BA
UCB$L_RECORD            =  000000B0
UCB$L_SVAPTE            =  00000078
UCB$L_VCB               =  00000034
UCB$M_MS_FEF            =  00000001
UCB$M_MS_LBA            =  00000400
UCB$M_MS_RDPR           =  00000020
UCB$M_MS_RPI            =  00000800
```

```
UCB$M_MS_SWAP                   = 00000002
UCB$M_MS_SWE                    = 00000040
UCB$M_MS_VCK                    = 00001000
UCB$M_ONLINE                    = 00000010
UCB$M_VALID                     = 00000800
UCB$Q_MS_BUFSVAPTE                000000EC
UCB$Q_MS_TMP1                     000000E0
UCB$V_INT                       = 00000001
UCB$V_MS_FEF                    = 00000000
UCB$V_MS_LBA                    = 0000000A
UCB$V_MS_RDPR                   = 00000005
UCB$V_MS_RPI                    = 0000000B
UCB$V_MS_SWAP                   = 00000001
UCB$V_MS_SWE                    = 00000006
UCB$V_MS_VCK                    = 0000000C
UCB$V_POWER                     = 00000005
UCB$V_VALID                     = 0000000B
UCB$W_BCNT                      = 0000007E
UCB$W_BOFF                      = 0000007C
UCB$W_DEVBUFSIZ                 = 00000042
UCB$W_DEVSTS                    = 00000068
UCB$W_FUNC                      = 0000009A
UCB$W_MS_LNH                      000000FA
UCB$W_MS_MHD                      000000F8
UCB$W_MS_RBPC                     000000FC
UCB$W_MS_SPACNT                   000000B4
UCB$W_MS_TSBA                     000000C0
UCB$W_MS_TSPT3                    000000BE
UCB$W_MS_TSSR                     000000C2
UCB$W_MS_XC                       000000C4
UCB$W_MS_XSR0                     000000FE
UCB$W_MS_XSR1                     00000100
UCB$W_MS_XSR2                     00000102
UCB$W_MS_XSR3                     00000104
UCB$W_STS                       = 00000064
UNLOAD                            00000706 R      03
VA$S_VPN                        = 00000015
VA$V_VPN                        = 00000009
VEC$B_DATAPATH                  = 00000013
VEC$L_IDB                       = 00000008
VEC$L_UNITINIT                  = 00000018
VEC$W_MAPREG                    = 00000010
WCB$W_NMAP                      = 00000016
WRITECHAR                         00000507 R      03
WRITECHECK                        000003F1 R      03
WRITECHECKR                       000003F1 R      03
WRITEDATA                         000004A4 R      03
WRITERET                          000004F1 R      03
WRITESUBS                         000004FC R      03
WRTTMK                            00000698 R      03
WRTTMKR                           000006E5 R      03
XTC                               00000A2F R      03
XTC1                              00000A3C R      03
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000106 (262.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$$$105_PROLOGUE	00000086 (134.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
$$$115_DRIVER	00000D19 (3353.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	33	00:00:00.05	00:00:01.89
Command processing	120	00:00:00.39	00:00:03.56
Pass 1	655	00:00:22.16	00:01:14.43
Symbol table sort	0	00:00:02.80	00:00:14.50
Pass 2	410	00:00:05.50	00:00:22.91
Symbol table output	1	00:00:00.21	00:00:00.60
Psect synopsis output	0	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1221	00:00:31.12	00:01:57.91

The working set limit was 2550 pages.
180605 bytes (353 pages) of virtual memory were used to buffer the intermediate code.
There were 140 pages of symbol table space allocated to hold 2533 non-local and 147 local symbols.
2396 source lines were read in Pass 1, producing 27 object records in Pass 2.
51 pages of virtual memory were used to define 49 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	32
_255DUA28:[SYSLIB]STARLET.MLB;2	12
TOTALS (all libraries)	44

2529 GETS were required to define 44 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:TSDRIVER/OBJ=OBJ$:TSDRIVER MSRC$:TSDRIVER/UPDATE=(ENH$:TSDRIVER)+EXECML$/LIB

TUDRIVER
.LIS

XADRIVER
.LIS

TSDRIVER
.LIS